



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

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



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**LAB RECORD SOFTWARE
ENGINEERING LAB**

B.Tech. III YEAR I SEM (RKR21) ACADEMIC YEAR 2024-25



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



NBA

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)**



NBA
National Basketball Association

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

Daily Laboratory Assessment Sheet

Name of the Lab:

Name of the Student:

Class:

HT. No:

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

INDEX

S.NO	CONTENTS	PAGE NO
I	Vision/Mission /PEOs/POs/PSOs	
II	Syllabus	
III	Course outcomes, C0-PO Mapping	
Exp No:	List of Experiments	
1	INSTALLATION OF STARUML, GIT BASH AND GITHUB ACCOUNT CREATION	
2	IEEE – SRS (SOFTWARE REQUIREMENTS SPECIFICATION) WITH UML DIAGRAMS	
3	INSTALLATION OF ECLIPSE, MAVEN, JDK, TOMCAT, CONFIGURING TOMCAT TO ECLIPSE	
4	A. BASIC GIT COMMANDS - VERSION, CONFIG, INIT, STATUS, ADD, COMMIT, DIFF, HELP B. GIT COMMANDS: WORKING WITH LOCAL AND REMOTE REPOSITORIES - BRANCHES, CHECKOUT, MERGE, REVERT, LOG C. GIT COMMANDS: WORKING WITH REMOTE REPOSITORIES - REMOTE, CLONE, PULL, PUSH, FORK	
5	A. CREATING MAVEN JAVA PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB. B. CREATING MAVEN WEB PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB. C. INSTALLATION OF JENKINS.	
6	A. BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT B. BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN WEB PROJECT WITH POLL SCM C. BUILDING THE CI/CD SCRIPTED PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT WITH POLL SCM	
7	A. INSTALLATION OF DOCKER, MINIKUBE, ACCOUNT IN DOCKERHUB B. DOCKER CLI COMMANDS C. MODIFY AND PUSH DOCKER IMAGE D. CREATE AND PUSH DOCKERFILE IMAGE E. RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE F. DEPLOYING AND SCALING APPLICATIONS USING MINIKUBE G. DEPLOYING AND MANAGING MONITORING SYSTEMS USING NAGIOS IN DOCKER	

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



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering

Vision of the 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



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

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



Department of Computer Science & Engineering

Vision of the Department:

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

Mission of the Department:

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



Department of Computer Science & Engineering

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. **PO2: Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

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

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

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

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

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

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

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

PO10: Communication: Communicate effectively on complex engineering activities with the

engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

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

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



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering

PROGRAM SPECIFIC OUTCOMES (PSOs)

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

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



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

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



Department of Computer Science & Engineering

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will 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

Pre-requisites/ Co-requisites:

0 0 3 1.5

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.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering 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

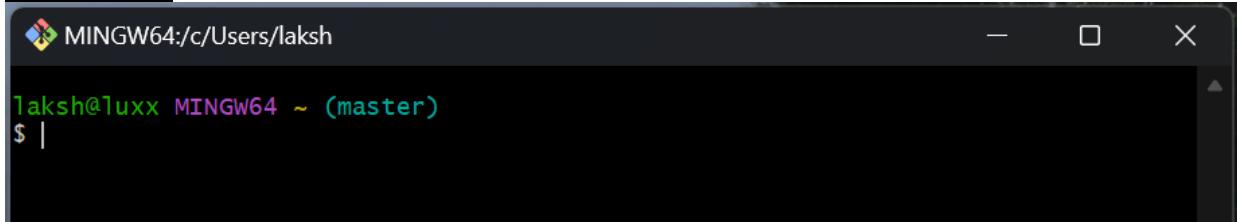
Experiment 1:

INSTALLATION OF STARUML, GIT BASH AND GITHUB ACCOUNT CREATION

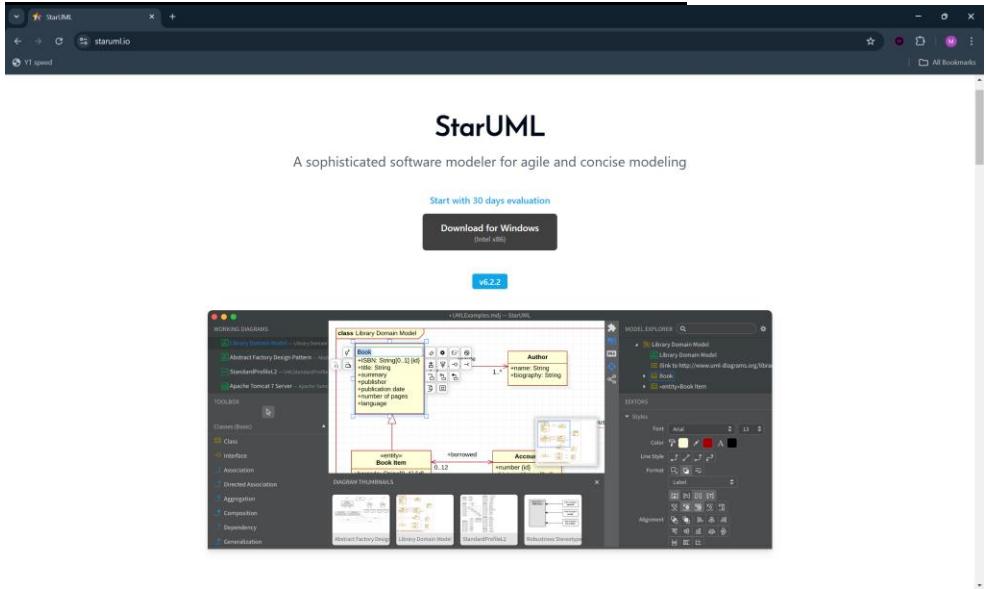
GIT BASH INSTALLATION:



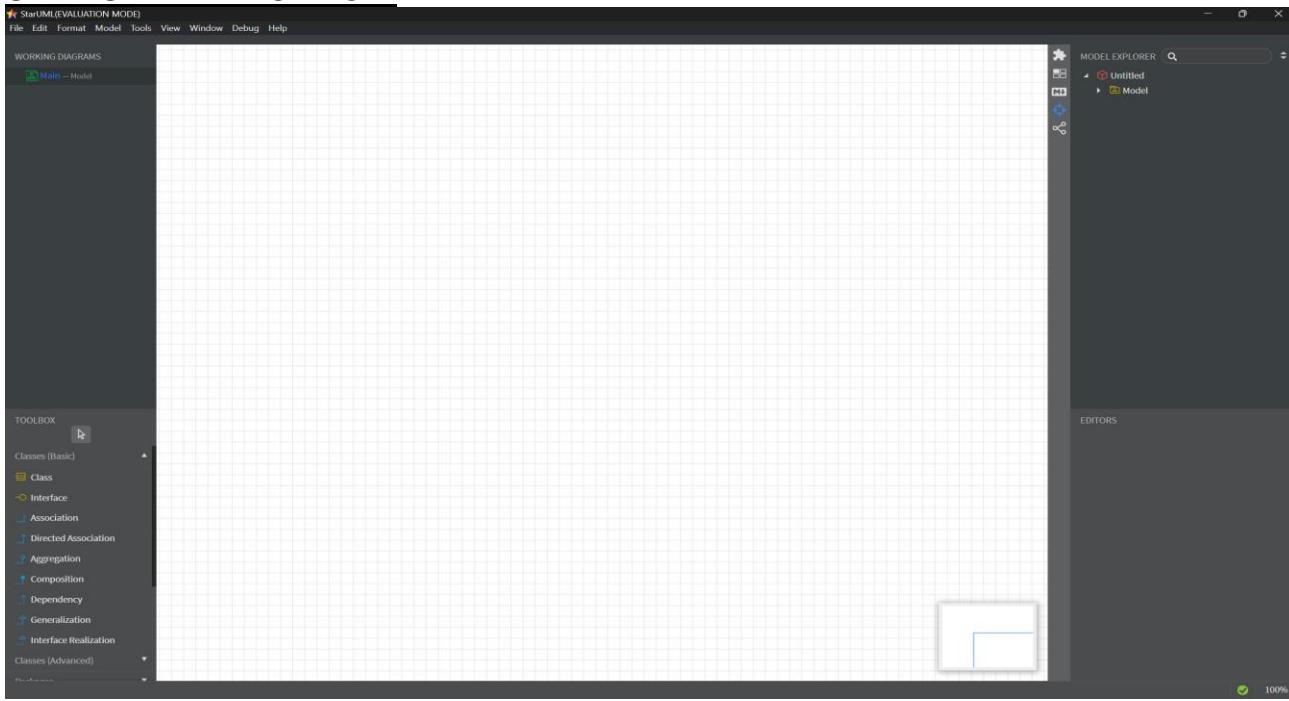
GIT BASH :



STAR UML DOWNLOAD AND INSTALLATION:



STAR UML APPLICATION:



GITHUB ACCOUNT:

A screenshot of a GitHub user profile for the account "lkksh". The profile picture is a circular logo with a red and white pixelated pattern. The top navigation bar shows the user's name "lkksh" and links for Overview, Repositories (5), Projects, Packages, and Stars. A message box says "You unlocked new Achievements with private contributions! Show them off by including private contributions in your Profile in settings." Below the message, there's a section for "Popular repositories" featuring "face-recognition" (Public, C), "SE_example" (Public), "simple" (Public, Forked from orf/simple, CSS), "maven" (Public, Java), and "WebProject" (Public, Java). On the left, there's an "Achievements" section with a blue circular icon. At the bottom, it shows "23 contributions in the last year" and "Contribution settings". The date "2025" is highlighted in a blue box.

Experiment 2:

IEEE – SRS (SOFTWARE REQUIREMENTS SPECIFICATION) WITH UML DIAGRAMS

CourseTide

- | | | |
|---------------------------|---|------------|
| 1. Mohammed Abdur Rasheed | - | 22BD1A0538 |
| 2. Nehith Sayini | - | 22BD1A053F |
| 3. Laksh Vijayvargiya | - | 22BD1A050X |
| 4. Jyothi Swaroop | - | 22BD1A050D |

Problem Statement:

Online education has become a crucial part of learning, offering students flexibility and convenience. However, many course reservation platforms focus only on enabling users to enroll in courses, neglecting critical aspects

such as peer collaboration and personalized support. This lack of engagement and guidance can hinder students' overall learning experience, as they often struggle to find study partners or receive real-time assistance with course-related queries. Course Tide aims to address these issues by providing a unique, feature-rich online course reservation system that emphasizes both collaborative learning and interactive support.

One of the primary challenges students face in existing systems is the inability to engage with peers who are enrolled in the same course. Collaborative learning is a proven method for enhancing knowledge retention and motivation, yet many platforms fail to facilitate group interaction. Course Tide's Collaborative Learning Spaces provide a solution by allowing students to form study groups, participate in forums, and collaborate on course-related discussions. This feature encourages knowledge sharing, group problem-solving, and helps students stay connected throughout the course duration. It also enables the exchange of ideas and resources, making learning more dynamic and engaging.

Another key issue is the lack of personalized, real-time support for students navigating the platform or managing their course selections. Many users have difficulty choosing the right course or need quick help with specific queries, yet traditional course systems do not provide an efficient way to address these needs. To solve this,

Course Tide will integrate an AI-powered Interactive Course Chatbot. This chatbot will assist users with personalized course recommendations based on their learning history, preferences, and career goals. Additionally, it will provide real-time support for course-related questions, technical issues, and scheduling, helping students make informed decisions and resolve issues quickly.

By addressing both collaboration and personalized assistance, Course Tide sets itself apart from existing platforms that focus solely on

course enrolment. The combination of Collaborative Learning Spaces and the Interactive Course Chatbot ensures that students not only reserve courses but also engage deeply with the learning process.

The platform is designed to be more than just a booking system; it offers a holistic learning experience that supports students from enrolment to course completion.

Moreover, Course Tide will offer a user-friendly, responsive interface that allows students to easily navigate through course options, interact with peers, and access chatbot assistance seamlessly. The system will also provide real-time notifications, ensuring that students are updated on their course progress, upcoming sessions, and any collaborative group activity. This will keep users engaged and help them stay on track throughout their educational journey.

In conclusion, Course Tide is designed to enhance the traditional online course reservation system by integrating essential features for collaboration and personalized support. The Collaborative Learning Spaces foster peer interaction and teamwork, while the Interactive Course Chatbot ensures students receive the guidance they need when navigating the platform. By offering these features, Course Tide aims to provide a more interactive, engaging, and supportive learning environment that caters to the needs of modern students.

Software Requirement Specifications

For

Course Tide Website

Version 1.0 approved

Prepared by:

1. Mohammed Abdur Rasheed - 22BD1A0538
2. Nehith Sayini - 22BD1A053F
3. Laksh Vijayvargiya - 22BD1A050X
4. Jyoti Swaroop - 22BD1A0553

05-09-2024.

Table of Contents

Table of Contents.....	Error! Bookmark not defined.	
Revision History	Error! Bookmark not defined.	
1. Introduction	Error! Bookmark not defined.	
1.1	Purpose	Error!
 Bookmark not defined.		
1.2	Document Conventions	Error!
 Bookmark not defined.		
1.3	Intended Audience and Reading Suggestions	Error!
 Bookmark not defined.		
1.4	Product Scope	Error!
 Bookmark not defined.		
1.5	References	Error!
 Bookmark not defined.		
2. Overall Description	Error! Bookmark not defined.	
2.1	Product Perspective	Error!
 Bookmark not defined.		
2.2	Product Functions	Error!
 Bookmark not defined.		
2.3	User Classes and Characteristics	Error!
 Bookmark not defined.		
2.4	Operating Environment	Error!
 Bookmark not defined.		
2.5	Design and Implementation Constraints	Error!
 Bookmark not defined.		
2.6	User Documentation	Error!
 Bookmark not defined.		
2.7	Assumptions and Dependencies	Error!
 Bookmark not defined.		
3. External Interface Requirements.....	Error! Bookmark not defined.	
3.1	User Interfaces	Error!
 Bookmark not defined.		

3.2	Hardware Interfaces	Error!
Bookmark not defined.		
3.3	Software Interfaces	Error!
Bookmark not defined.		
3.4	Communications Interfaces	Error!
Bookmark not defined.		
4. System Features	Error! Bookmark not defined.	
4.1	System Feature 1	Error!
Bookmark not defined.		
4.2	System Feature 2 (and so on)	Error!
Bookmark not defined.		
5. Other Nonfunctional Requirements	Error! Bookmark not defined.	
5.1	Performance Requirements	Error!
Bookmark not defined.		
5.2	Safety Requirements	Error!
Bookmark not defined.		
5.3	Security Requirements	Error!
Bookmark not defined.		
5.4	Software Quality Attributes	Error!
Bookmark not defined.		
5.5	Business Rules	Error!
Bookmark not defined.		
6. Other Requirements	5	

Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason for changes	Version
Week-1	05-09-2024	SRS creation(Introduction)	1.0
Week-2	06-09-2024	SRS updation(overall description)	2.0

Week-3	13-09-2024	SRS updation (external interface requirements)	3.0
Week-4	20-09-2024	SRS updation(system features and other non-functional requirements)	4.0
Week - 5	27-09-2024	SRS updating (SDLC)	5.0
Week - 6	18-10-2024	SRS updation (Use case diagrams)	6.0

1. Introduction

The CourseTide project aims to streamline the process of enrolling in educational courses by providing a user-friendly platform. This system is designed to simplify course registration, manage scheduling conflicts, and facilitate communication between students and instructors. The introduction will cover the background of the project, its significance in the educational sector, and an overview of the features and functionalities of the reservation system.

1.1 Purpose

The primary purpose of this project is to create a comprehensive, efficient, and reliable course reservation system that enhances the enrolment experience for both students and administrators. It will address the challenges associated with manual registration processes, such as scheduling conflicts, overbooked classes, and administrative inefficiencies. By automating these processes, the system aims to reduce errors, save time, and provide a seamless user experience.

1.2 Document Convention

This document adheres to standard conventions to ensure clarity and consistency. Key conventions include:

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 is intended for a diverse audience including:

Project Stakeholders: Individuals involved in funding, approving, or overseeing the project.

Developers: Technical teams responsible for designing, coding, and implementing the system.

End Users: Students and administrative staff who will interact with the reservation system.

Documentation Team: Authors and reviewers who are responsible for maintaining and updating the project documentation.

For effective reading: Project Stakeholders should focus on sections detailing the purpose and product scope to understand the

project's value and objectives. Developers should pay close attention to the technical requirements and system specifications. End Users might benefit from sections describing system features and user interface guidelines. Documentation Team should review the document conventions and references to ensure consistency and accuracy.

1.4 Product Scope

The product scope outlines the boundaries and deliverables of the course reservation system. This includes:

Core Features: Online course search, enrolment, scheduling, and payment processing.

User Roles: Different functionalities for students, instructors, and administrative staff.

System Requirements: Technical specifications, hardware, and software prerequisites.

Limitations: Any constraints or exclusions, such as integration with third-party systems or support for specific types of courses.

The scope defines what will be included in the project and helps manage expectations by specifying what is outside the project's boundaries.

1.5 References:

We took references from various websites related to online learning platforms like Coursera, Udemy, edX, Khan Academy, and LinkedIn Learning.

- Udemy: <https://www.udemy.com/blog/write-software-requirements-specifications/>
- edX: <https://www.edx.org/course/software-requirements-analysis>
- Khan Academy: <https://www.khanacademy.org/computing/computer-programming>
- LinkedIn Learning: <https://www.linkedin.com/learning/topics/software-requirements>
- Coursera: <https://www.coursera.org/articles/how-to-write-software-requirements-specifications>

Additionally, we referred to relevant documents and software requirements guidelines:

- Ch9: Software Requirements Document Guidelines:
http://users.csc.calpoly.edu/~csturner/courses/308w09/Ch9_DocReqts.pdf
- Software Requirements Specification for FSoft_D:
https://www.academia.edu/8831452/Software_Requirements_Specification_for_FSoft_D

References of articles:

- Udemy vs Coursera vs edX: The Ultimate Comparison (2023): <https://www.transizion.com/udemy-vs-coursera-vs-edx/>
- SRS Documents: Requirements and Diagrammatic Notations on Coursera (2023):
<https://www.coursera.org/articles/software-requirements-specifications-srs-diagrams>

2. Overall Description

2.1 Product Perspective

Course Tide is an innovative online course reservation system designed to improve upon existing online learning platforms like Coursera, Udemy, and edX. While it incorporates many of the standard features of these platforms, such as course browsing, enrolment and user account management, Course Tide stand out with its Collaborative Learning Spaces and Interactive Course

Chatbots. These features encourage peer interaction and offer personalized support, enhancing the overall learning experience. The platform is user-friendly and mobile-responsive, allowing students to access courses and engage with peers from any device. It integrates modern design with intuitive navigation and provides personalized learning suggestions, real-time support, and collaboration tools. Course Tide seeks to bridge the gap between passive course registration and an engaging, interactive learning environment.

2.2 Product Functions

2.2.1 Administrator

- Administrators have the ability to add, modify, and delete courses.
- Manage user accounts by approving or rejecting instructor or student registrations based on platform policies.
- Administrators can organize and categorize courses, ensuring they are easy to browse.
- Set up system notifications for course discounts, promotions, or group learning sessions.
- Monitor course completion rates and handle feedback or issues reported by users.

2.2.2 Instructors

- Instructors can create, update, and delete courses, manage course content, and upload learning materials.
- They can interact with students through forums and collaborative learning groups, enhancing engagement.
- Can track student progress and provide feedback on assessments or assignments.

2.2.3 Students/Users

- Students can search and enroll in courses based on subject, difficulty level, instructor, or specific skills.
- They have the ability to form study groups and engage in Collaborative Learning Spaces with peers enrolled in the same course.
- Can access personalized course recommendations and real-time assistance through Interactive Course Chatbots.
- Students are provided with account management features, such as tracking course progress, earning certificates, and participating in forums or discussions.

2.3 Operating Environment

- Course Tide is a web-based platform compatible with all modern browsers such as Google Chrome, Firefox, Safari, and Edge.
- The platform is fully responsive and optimized for both desktop and mobile devices.
- It requires a minimum 1 GHz processor, 2 GB RAM, and 512 MB available disk space for smooth operation on desktop devices.

- For mobile users, it supports iOS, Android, and Windows operating systems.

2.4 User Characteristics

The primary users of Course Tide are students, instructors, and administrators.

- Students are expected to have a basic understanding of computers and mobile devices to browse, enrol in, and complete courses.
- Instructors should have proficiency in managing course materials, interacting with students through the collaborative tools, and monitoring student progress.
- Administrators must be well-versed in platform operations, troubleshooting, and course or user management.

2.5 Design and Implementation Constraints

- Users must have a stable internet connection to access the platform and participate in live group discussions or collaborative spaces.
- All user data, course progress, and collaboration history will be stored securely in a cloud-based database.
- Course Tide will be available 24/7, ensuring students can access their enrolled courses and interactive features at any time.
- No specific operating system is required; the platform is platform-agnostic, running on any device with internet browsing capabilities.
- Users must use valid credentials to access their accounts, and two-factor authentication is recommended for additional security.

2.6 Assumptions and Dependencies

- Course Tide requires integration with third-party services such as video streaming (e.g., YouTube, Vimeo) for course content delivery and cloud storage solutions for storing course materials.
- The success of Course Tide depends on the availability of internet access and the familiarity of users with online platforms.
- The platform assumes instructors and students are comfortable using discussion forums, collaborative tools, and chatbot interactions for enhanced learning experiences.
- Continuous internet access and updated browsers are required for seamless platform use.

3. External Interface Requirements

3.1 User Interfaces

1. Course Search and Filtering UI

- **Goal:** Create a straightforward, intuitive search function.
- **Features:**

- **Search by Keyword:** A search bar for entering course-related keywords.
- **Filter Options:** Drop-down menus or checkboxes for:
 - Difficulty Level (e.g., Beginner, Intermediate, Advanced)
 - Course Type (e.g., Online, Onsite, Hybrid)
 - Certification (e.g., with or without certificate)
 - Start Date (selectable date ranges)
- **Responsiveness:** Ensure the interface works smoothly on both desktop and mobile devices.

2. Course Details Page

- **Goal:** Provide comprehensive course information to the user.
- **Features:**
 - Title, Instructor Name, Duration, Delivery Format (e.g., video, written), Difficulty Level
 - Pricing details
 - Additional course information (e.g., syllabus, prerequisites, reviews)
 - Buttons for enrolling in or saving the course for later.

3. Course Summarization and Celebrity Voice Features

- **Goal:** Allow users to customize the course experience through AI summarization and voice narration.
- **Features:**
 - **Summarization Options:** Users can select specific sections of the course to summarize.
 - **Celebrity Voice Selection:** A drop-down list of available celebrity voices for text-to-speech (TTS) narration.
 - **Audio Playback:** An embedded audio player for listening to the summarized content.

4. User Registration/Login Interface

- **Goal:** Provide secure and seamless user authentication.
- **Features:**
 - **Registration:** Fields for name, email, password, and possibly two-factor authentication (2FA) setup.
 - **Login:** Secure authentication with password or social media login options.

- **Profile Management:** Manage enrolled courses, personalized recommendations, and track learning progress.

3.2 Hardware Interfaces

1. Client Devices

- The platform should be optimized to run smoothly on various devices, including:
 - **Desktops and Laptops:** Fully responsive web interface.
 - **Tablets and Smartphones:** A mobile-friendly or adaptive design for smaller screens.

2. Servers

- The platform will need cloud-based servers to handle:
 - **Web Hosting:** The platform's frontend and backend services.
 - **Databases:** Storing user data, course information, and transaction records.
 - **Course Recommendation Engine:** AI-based algorithm for course suggestions.
 - **AI-Powered Summarization:** Hosting and executing machine learning models.
 - **Text-to-Speech (TTS):** Servers capable of processing and delivering narrated content.

3.3 Software Interfaces

1. Text-to-Speech (TTS) API

- Integration with services like Google Cloud TTS, Amazon Polly, or Eleven Labs to generate course audio using selected celebrity voices.

2. AI Summarization API

- Use AI models from providers like OpenAI or Hugging Face to summarize course materials, such
 - as PDFs and video content.

3. Payment Gateway API

- Implement services like Stripe or PayPal to handle secure payments for course enrolments.

4. Email and Notification API

- Connect with an email service to send confirmations, reminders, and summary notifications to users.

3.4 Communications Interfaces

1. HTTPS

- Secure communication between the platform and client devices via HTTPS for encrypted data transfer.

2. REST API

- REST APIs will handle data exchanges between the frontend, backend, and third-party services like TTS and summarization.

4. System Features

4.1 System 1: Course Search, Filter, and Enrolment

- **Description:** Users can search for courses using keywords and apply filters based on course type, difficulty level, certification, and pricing. The system updates results dynamically based on user selections.
- **Priority:** High
- **Stimulus/Response Sequences:**
 - User enters a keyword and applies filters.
 - The system displays relevant courses.
 - User selects a course to view details and can proceed with enrollment.
- **Functional Requirements:**
 - **F1.1:** Support keyword-based searches.
 - **F1.2:** Implement filtering options.
 - **F1.3:** Provide course details and enrolment functionality.

4.2 System Feature 2: Celebrity Voice Course Narration and AI Summarization

- **Description:** Users can select a celebrity voice for course narration and request AI-generated summaries of specific course sections, whether PDF documents or videos.
- **Priority:** High
- **Stimulus/Response Sequences:**
 - User chooses a course section, selects a celebrity voice, and the system generates an audio narration.
 - User uploads a PDF or selects a video, and the system provides a summary.
- **Functional Requirements:**
 - **F2.1:** Integrate with TTS service to offer celebrity voices.
 - **F2.2:** Enable AI-powered summarization for course materials.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system must handle large numbers of concurrent users.
- Search results and summaries should be delivered within 2-3 seconds.
- TTS generation should complete within 5-10 seconds for standard-length course sections.

5.2 Safety Requirements

- Ensure data integrity, protecting user data such as course history and payment details.
- Users should be able to retrieve their progress in case of system failure.

5.3 Security Requirements

- Secure user authentication using encryption (SSL/TLS) and multi-factor authentication (MFA).
- Payment processing must comply with PCI DSS standards.

5.4 Software Quality Attributes

- **Usability:** Intuitive, engaging user experience.
- **Scalability:** Support growing user demand, especially during peak periods.
- **Reliability:** Ensure 99.9% uptime.
- **Maintainability:** Modular, well-documented codebase for future updates.

5.5 Business Rules

- The platform will charge a commission fee for each course enrolment.

Software Development Life Cycle (SDLC)

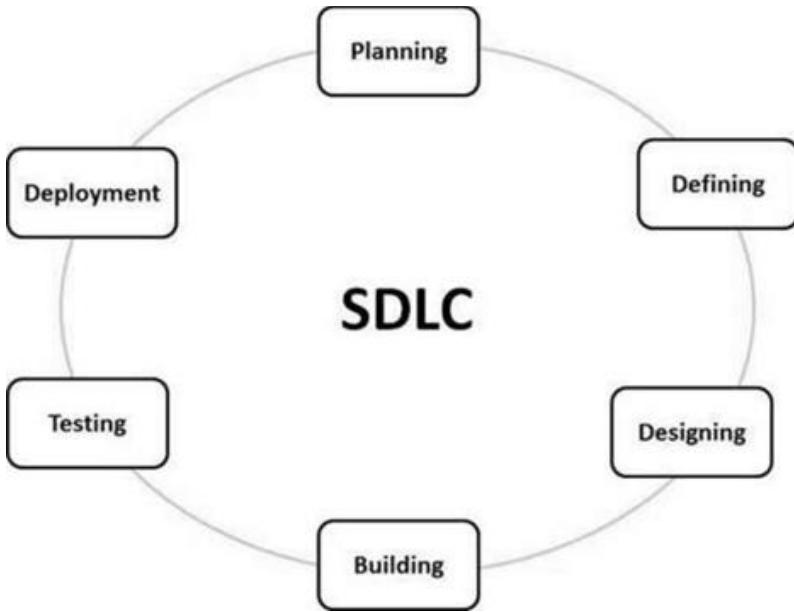
Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called the Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters such as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation

with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred to as Software Development Process Models".

Each process model follows a series of steps unique to its type to ensure success in the process of software development. Following are the most important and popular SDLC models followed in the industry –

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

1. Introduction

1.1 Purpose of Document

Provide an introductory paragraph explaining the purpose of this document. Its purpose is to explicitly cite all functions that the project shall do. This document is the primary document, upon which the design, source code, and test plan all base their content. This document is used to determine if the final delivered product provides everything that it was supposed to. The Client, User, and Software Engineering representatives often negotiate the content of this document.

1.2 Scope

Provide two paragraphs, the first describing the scope of the product, with the second describing the scope of this document. Remember that "scope" basically means the extent of activity or influence, or range of operation. Be sure that the two paragraphs in this section distinguish between the scope of the product, versus the scope of this document. You will probably find that in most of the Software Engineering documents that you create in this course, the paragraph for scope of product will be identical (as expected). Specifically for this document, the scope includes all team members and their responsibilities for specifying the product's requirements.

1.3 Objective

A project objective describes the desired results of a project, which often includes a tangible item. An objective is specific and measurable, and must meet time, budget, and quality constraints. ... A project may have one objective, many parallel objectives, or several objectives that must be achieved sequentially.

1.4 Proposed System

The proposed system should have the following features. The transactions should take place in a secured format between various clients in the network. It provides flexibility to the user to transfer the data through the network very easily by compressing the large amount of file.

2. Requirements Specifications

2.1 Functional Requirements

functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all

the cases where the system uses the functional requirements, these are captured in use cases. Functional requirements are supported by non-functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>," while non-functional requirements take the form "system shall be <requirement>." The plan for implementing functional requirements is detailed in the system design, whereas *non-functional* requirements are detailed in the system architecture.

2.2 Non-Functional Requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, nonfunctional requirements are just as critical as functional Epics, Capabilities, Features, and Stories. They ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do not fulfill mandatory requirements imposed by regulatory or standards agencies. In some cases, non-compliance can cause significant legal issues (privacy, security, safety, to name a few).

2.3 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

2.4 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

3. Literature Survey

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

When you write a literature review in respect of your project, you have to write the researches made by various analysts - their methodology (which is basically their abstract) and the conclusions they have arrived at. You should also give an account of how this research has influenced your thesis.

Descriptive papers may or may not contain reviews, but analytical papers will contain reviews. A literature review must contain at least 5 - 7 published researches in your field of interest.

4. System Designing

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements

to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

Diagrams in the UML

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

1. **Activity Diagrams** – We use Activity Diagrams to illustrate the flow of control in a system. We can also use an activity diagram to refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on the condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.
2. **Use Case Diagrams** – Use Case Diagrams are used to depict the functionality of a system or a part of a system. They are widely used to illustrate the functional requirements of the system and its interaction with external agents(actors). A use case is basically a diagram representing different scenarios where the system can be used. A use case diagram gives us a high-level view of what the system or a part of the system does without going into implementation details.
3. **Sequence Diagram** – A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios
4. to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.
5. **Class Diagram** – The most widely used UML diagram is the class diagram. It is the building block of all object-oriented software systems. We use class diagrams to depict the static structure of a system by showing the system's classes, their methods and attributes. Class diagrams also help us identify relationships between different classes or objects.

5. Implementation

The software implementation stage involves the transformation of the software technical data package (TDP) into one or more fabricated, integrated, and tested software configuration items that are ready for software acceptance testing. The primary activities of software implementation include the:

- Fabrication of software units to satisfy structural unit specifications.
- Assembly, integration, and testing of software components into a software configuration item.
- Prototyping challenging software components to resolve implementation risks or establish a fabrication proof of concept.
- Dry-run acceptance testing procedures to ensure that the procedures are properly delineated and that the software product (software configuration items (CIs and computing environment) is ready for acceptance testing.

6. Testing

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises Validation and Verification.

Software Validation

Validation is the process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – "Are we developing the product which attempts all that user needs from this software?".
- Validation emphasizes on user requirements.

Software Verification

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

- Verification ensures the product being developed is according to design specifications.
- Verification answers the question– "Are we developing this product by firmly following all design specifications?"
- Verifications concentrate on the design and system specifications.

7.Conclusion

SRS helps the customers to define their needs with accuracy, while it helps the development team understand what the customers need in terms of development. Investing time in writing the SRS document will lead to the successful development of the software the customer needs.

SOFTWARE REQUIREMENTS

Functional Requirements:

- These are statements of services the system should provide
 - =>how the system should react to particular inputs and
 - =>how the system should behave in particular situations
- In some cases, the functional requirements may also explicitly state
 - => What the system should not do
- The functional requirements definition of a system should be both
 - => Complete [i.e. It means that all services required by the user should be defined]
 - => Consistent [i.e. it means that requirements should not have contradictory definitions]

Non- Functional Requirements:

- These are constraints on the services (Or) functions offered by the system
- They include

- => Timing Constraints
- => Constraint on development process
- => Standards and so on...
- Some non-functional requirements may be process rather than product requirements
- Customer imposes these process requirements for two reasons:
 - => System Quality
 - => System Maintainability

Non-Functional Requirements Types:

Product Requirements Process Requirements External Requirements

(i) Product Requirements:

These requirements result from the need for the delivered product, to behave in a particular way

Example:

- Requirements on how fast the system must execute and how much memory it requires
- Reliability Requirements [i.e, acceptable failure rate]
- Portability Requirements

(ii) Organizational Requirements:

- These requirements are consequence of organizational policies and procedures

Example:

Implementation requirements such as programming language (Or) design method

used

- Delivery Requirements which specify when the product and its documentation to be Delivered

(iii) External Requirements:

- These requirements arise from factors external to the system and its development process

Example:

- Interoperability Requirements which specify how the system interacts with systems in other organizations
- Legislative Requirements, which ensure that the system operates within the law

An Overview of UML

Unified Modeling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

A Conceptual Model of UML

- A conceptual model can be defined as a model which is made of concepts and their relationships.
- A conceptual model is the first step before drawing a UML diagram. It helps to understand the entities in the real world and how they interact with each other.

As UML describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. The conceptual model of UML can be mastered by learning the following three major elements –

- UML building blocks
- Rules to connect the building blocks
- Common mechanisms of UML

Object Oriented Concepts Used in UML –

1. Class – A class defines the blueprint i.e. structure and functions of an object.

2. Objects – Objects help us to decompose large systems and help us to modularize our system. Modularity helps to divide our system into understandable components so that we can build our system piece by piece.

An object is the fundamental unit (building block) of a system which is used to depict an entity.

3. Inheritance – Inheritance is a mechanism by which child classes inherit the properties of their parent classes.

4. Abstraction – Mechanism by which implementation details are hidden from the user.

5. Encapsulation – Binding data together and protecting it from the outer world is referred to as encapsulation.

6. Polymorphism – Mechanism by which functions or entities are able to exist in different forms.

Diagrams in the UML

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

There are two broad categories of diagrams and they are again divided into subcategories –

1. Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

2. Behavior Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

Structural Diagrams

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are –

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

1. Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagrams consist of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature.

Active class is used in a class diagram to represent the concurrency of the system.

Class diagrams represent the object orientation of a system. Hence, it is generally used for development purposes. This is the most widely used diagram at the time of system construction.

2. Object Diagram

Object diagrams can be described as an instance of class diagrams. Thus, these diagrams are closer to real-life scenarios where we implement a system. Object diagrams are a set of objects and their relationship is just like

class diagrams. They also represent the static view of the system. The usage of object diagrams is similar to class diagrams but they are used to build a prototype of a system from a practical perspective.

3. Component Diagram

Component diagrams represent a set of components and their relationships. These components consist of classes, interfaces, or

collaborations. Component diagrams represent the implementation view of a system.

During the design phase, software artifacts (classes, interfaces, etc.) of a system are arranged in different groups depending upon their relationship. Now, these groups are known as components. Finally, it can be said component diagrams are used to visualize the implementation.

4.Deployment Diagram

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed. Deployment diagrams are used for visualizing the deployment view of a system. This is generally used by the deployment team.

Behavioral Diagrams:Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered. Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams –

- Use case diagram
- Sequence diagram
- Collaboration diagram
- Statechart diagram
- Activity diagram

1.Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, a use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

2.Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

3.Collaboration Diagram

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.

The purpose of the collaboration diagram is similar to a sequence diagram. However, the specific purpose of collaboration diagrams is to

visualize the organization of objects and their interaction.

4.Statechart Diagram

Any real-time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system.

Statechart diagram is used to represent the event driven state change of a system. It basically describes the state change of a class, interface, etc. State chart diagram is used to visualize the reaction of a system by internal/external factors

5.Activity Diagram

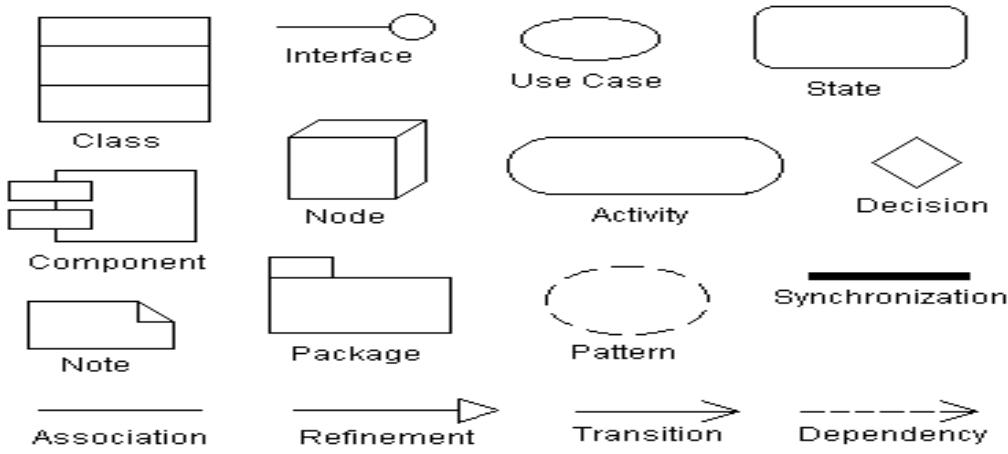
Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

Diagram Elements

Some of the graphical constructs from which diagrams are made are:

- Icon: graphical symbol of fixed size and shape (doesn't hold contents)
- Two-dimensional symbols: have variable size and can expand to hold contents, may be divided into compartments
- Paths: sequences of line segments with attached endpoints. The endpoints are always symbols (no dangling paths). May also have icons at the end to qualify the meaning of the path symbol.
- Strings: text
- Name: A string that uniquely identifies some model element within some scope
- Label: A string attached to a graphic symbol
- Keyword: Text enclosed within "«" and "»" to convey some concept. There are many keywords so we don't need zillions of specialized graphical symbols.
- Expression: A linguistic formula that yields a value
- Some model elements:



6. Other Requirements

- The system must comply with data privacy regulations, including **GDPR**, for managing user data.
- The platform should be localized to support **multiple languages** for a global user base.

Appendix A: Glossary

- TTS (Text-to-Speech)**: Technology that converts text into human-like speech.
- API (Application Programming Interface)**: A set of protocols for building and interacting with software applications.
- MFA (Multi-Factor Authentication)**: A security process requiring users to verify their identity using two or more verification methods.

Appendix B: Analysis Models

- UML Diagrams**: Use case diagrams to visualize user interactions with search, filter, celebrity voice, and summarization features.
- ER Diagrams**: Entity-Relationship diagrams representing data relationships between courses, users, and enrolments.

Appendix C: To Be Determined List

- Integration Partner for TTS API**: Final choice of TTS provider (Google, Amazon, or Eleven Labs) is pending.
- AI Summarization Model**: Decision between using OpenAI models or developing a custom summarization model is yet to be made.

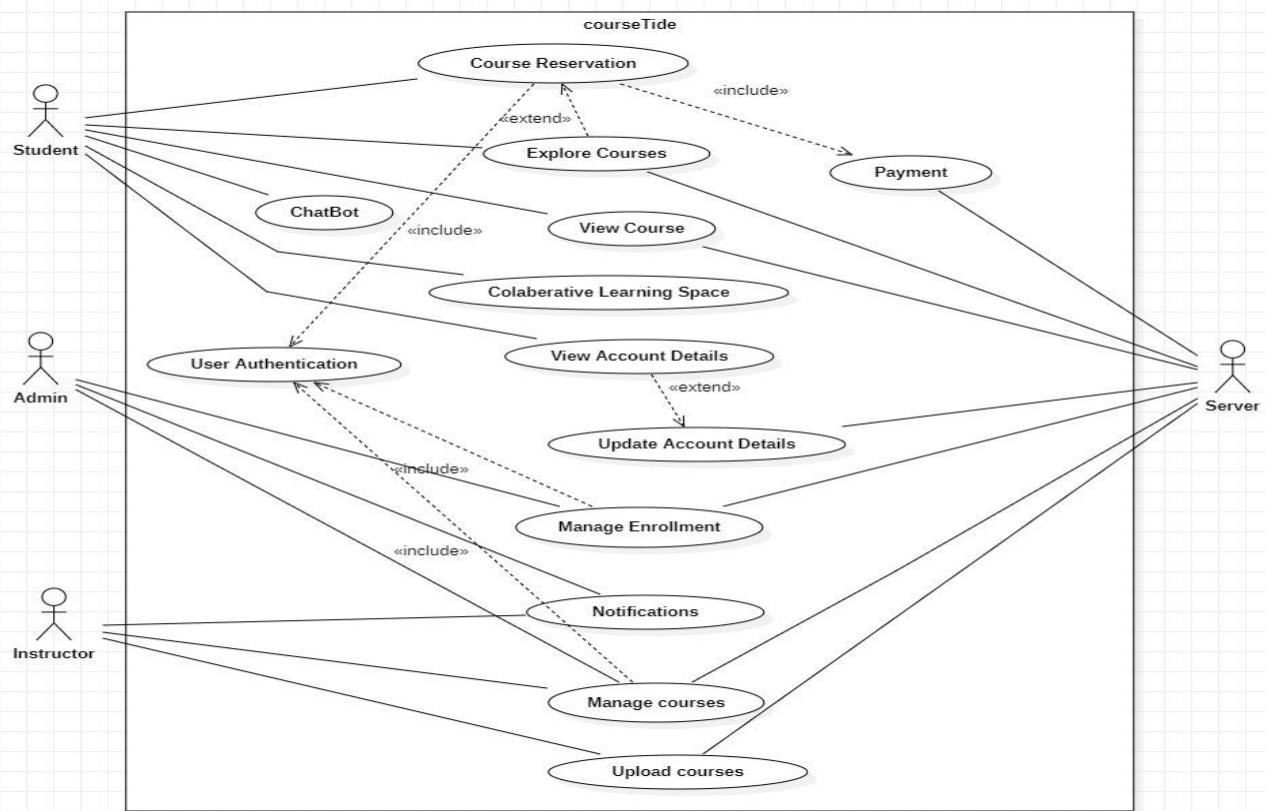
Use Case ID:	04566789999
Use Case Name:	CourseTide

End Objective:	Discovering, analysing and recommending courses based on user preferences		
Created by:	1.MohammedAbdurRasheed Nehith S 4.Jyothi Swaroop	2. 3. Laksh	On (date):
User/Actor:	Administrator, student, instructor, server		
Trigger:	The user (admin, student, instructor) attempts to access the system		
Basic/Normal Flows			
User Actions	System Actions		
<p>The user logs into the online course platform by entering valid credentials</p> <p>The user views their dashboard (Admin views course management, Instructor views enrolled students, Students view their enrolled courses).</p> <p>The user selects the View Course Content option</p> <p>The user edits their personal details or course information (if Admin or Instructor).</p> <p>The user enrolls in a course or creates a course (based on role).</p> <p>The user checks out and makes payments for course fees.</p>	<p>The system displays a login page prompting the user to provide the correct username and password.</p> <p>The system checks the user's role (Admin, Instructor, or Student) and displays relevant features (e.g., course management, student lists, or enrolled courses).</p> <p>The system retrieves and displays the course content associated with the user's enrolled courses.</p> <p>The system provides editable fields for personal or course information and updates the database with the new details after confirming the changes.</p> <p>The system presents enrollment or course creation options and confirms the action once submitted.</p> <p>The system processes the payment (using online methods) and updates the billing records, providing a receipt for the course enrollment or creation fees.</p>		

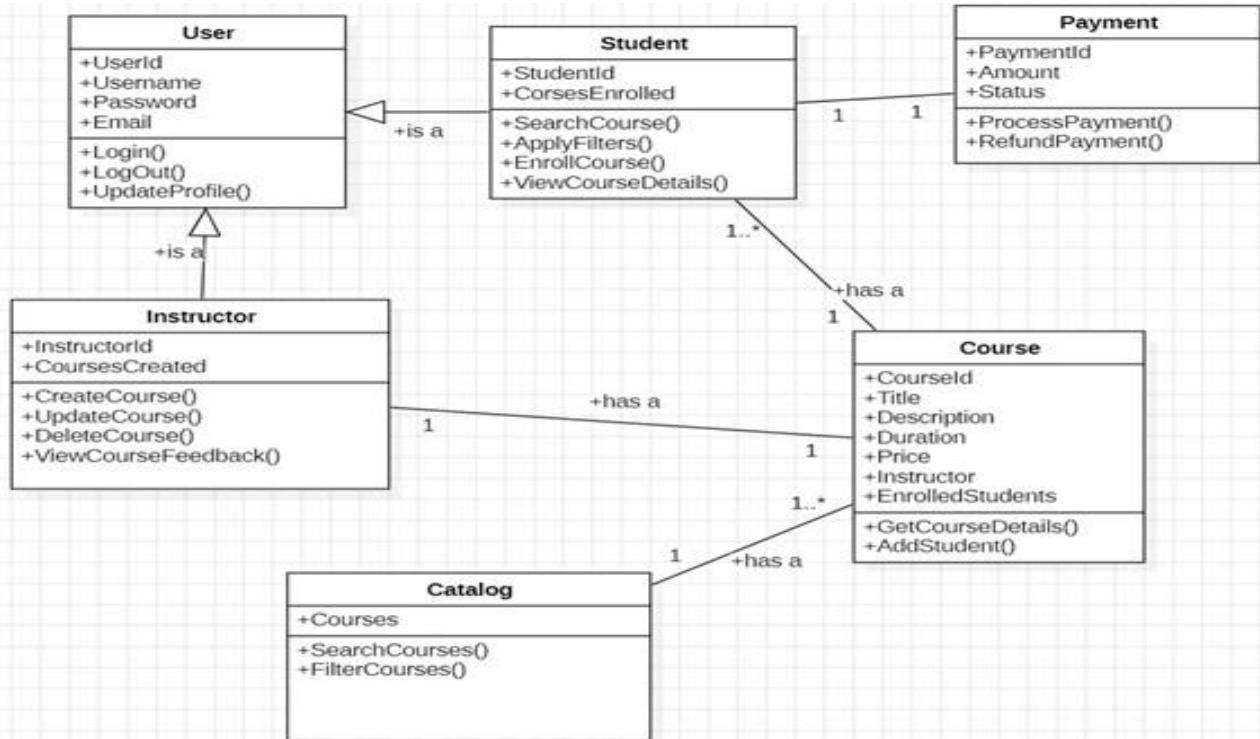
Exception Flows

User Actions	System Actions
User tries to login but doesn't have an account	The system requests the user to register an account on the registration page before allowing the login process.
User tries to login by entering incorrect details	The system displays an error message: "Please check the username or password entered" and prompts the user to re-enter correct login credentials.
User tries to view course content but lacks permission Instructor tries to create a course, but it has already been created.	The system shows a message: "Access denied. You do not have permission to view this data," and restricts access.
The student tries to enroll in a course, but no available seats	The system displays a message: "Restocking request already made. Please wait for the admin to process the current request."
Admin tries to update user data, but the data is corrupted	The system displays a message: No available seats. Please try again later or choose a different course The system displays an error message: "Unable to update data. Database error encountered. Please get in touch with support."

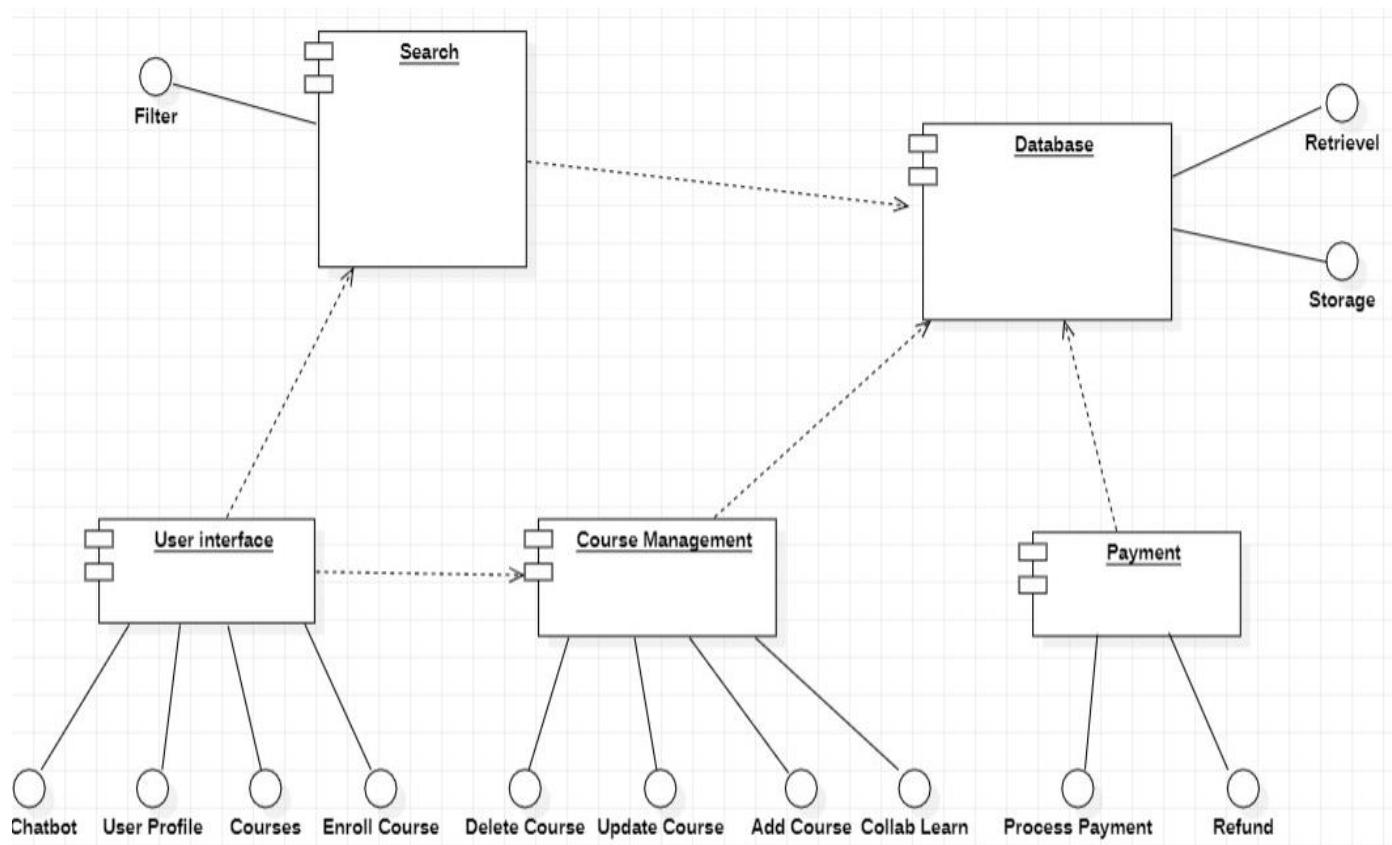
Use case diagram:



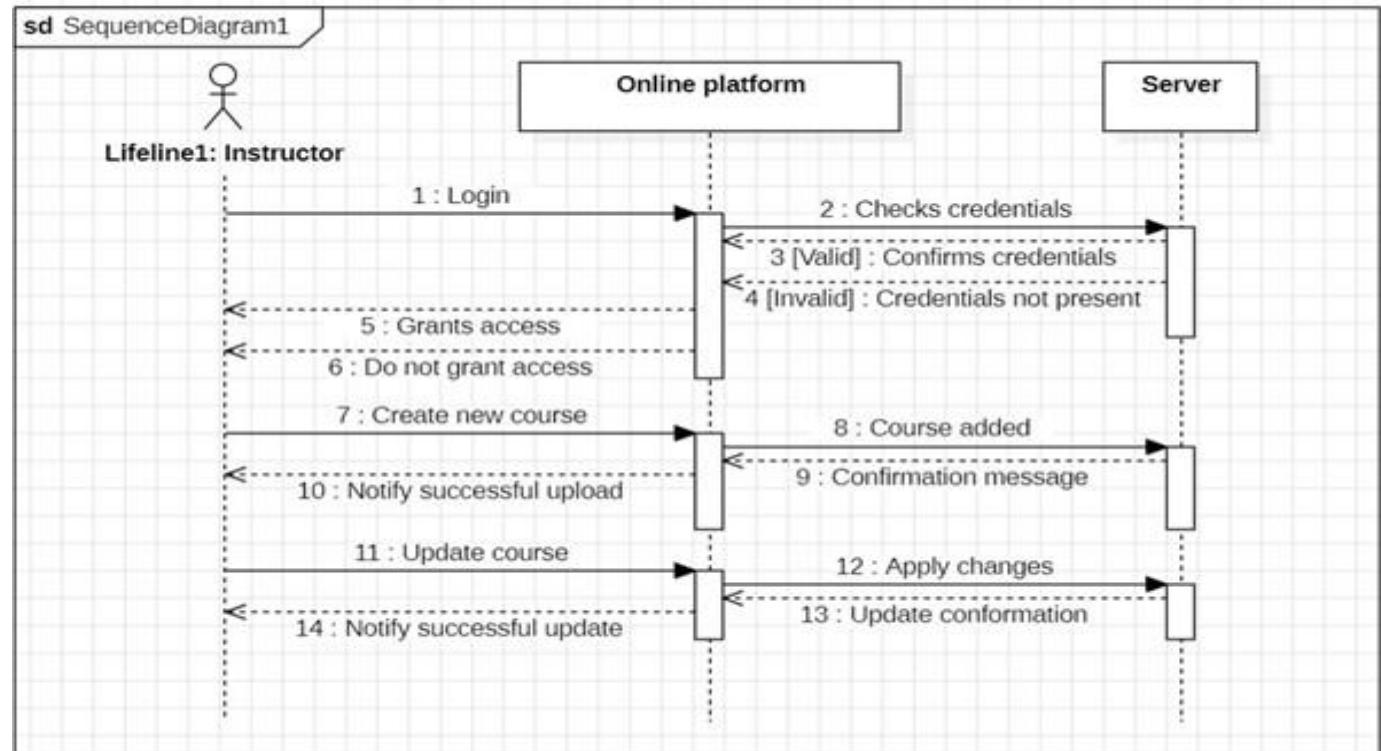
Class diagram:



Component diagram:



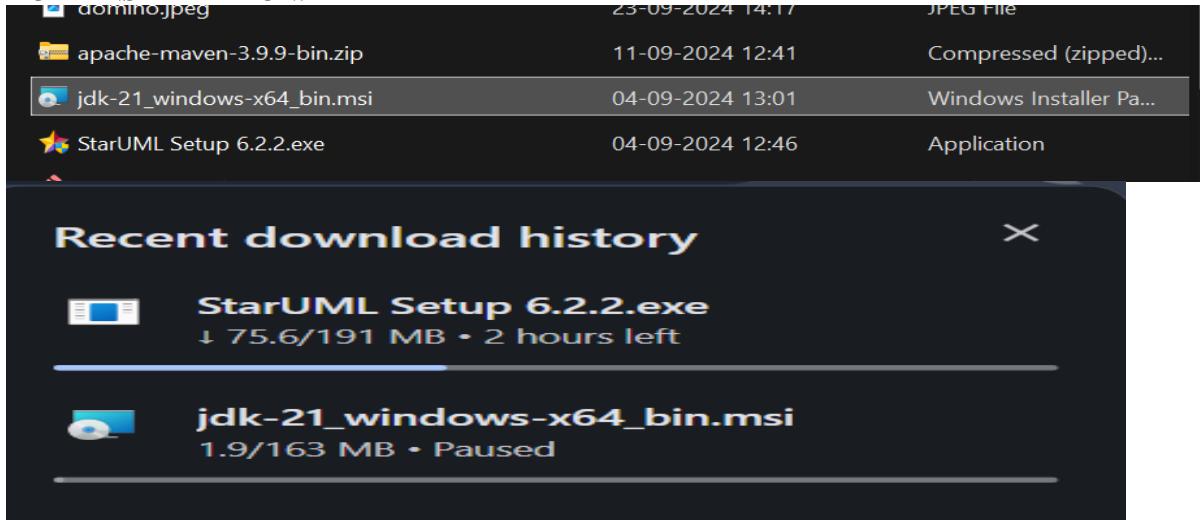
Sequence diagram:



Experiment 3:

INSTALLATION OF ECLIPSE, MAVEN, JDK, TOMCAT, CONFIGURING TOMCAT TO ECLIPSE

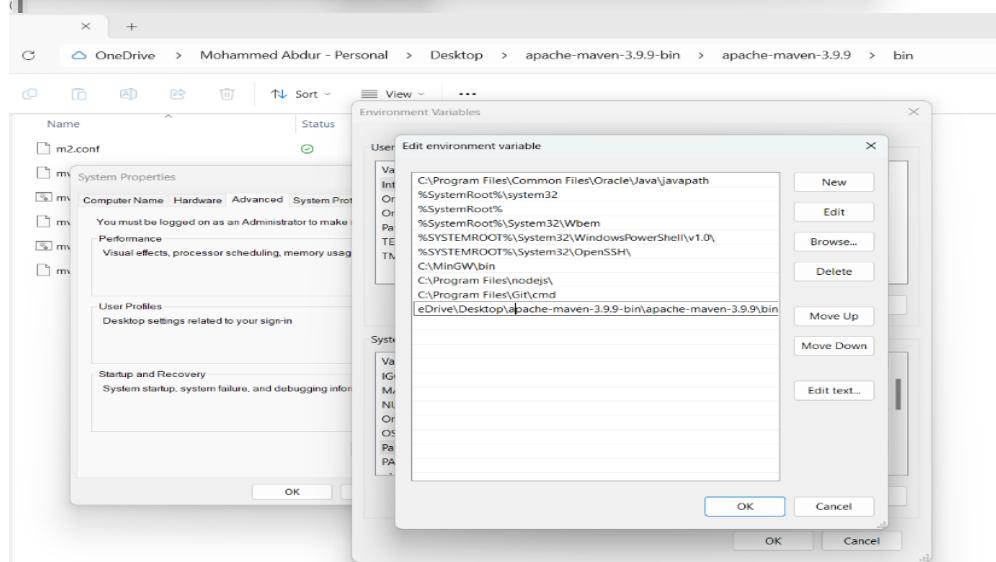
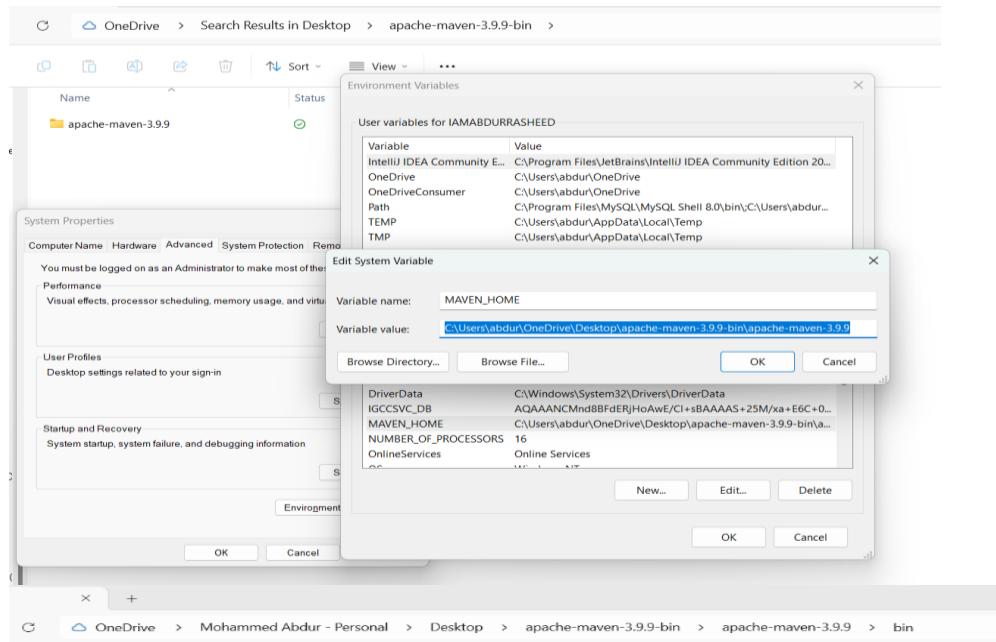
JDK INSTALLATION:



MAVEN INSTALLATION:

The image shows a screenshot of a web browser displaying the Apache Maven download page. The left sidebar contains links for Documentation, Plugins, Extensions, and various Maven Centres. The main content area shows release notes and details for the Maven installation itself, such as disk requirements (approximately 10MB) and operating system compatibility (no minimum requirement). It also provides a table of available Maven distributions (tar.gz, zip, source) with their respective links, checksums, and signatures. Below this, there's a section on Release Notes and another on Other Releases, which includes a note about using the latest version for the best features and bug fixes.

The bottom part of the image shows a Windows file explorer window titled 'Downloads' containing a folder named 'apache-maven-3.9.9-bin'. The file explorer interface includes a sidebar with navigation icons and a toolbar with file operations like New, Open, Save, and Print.

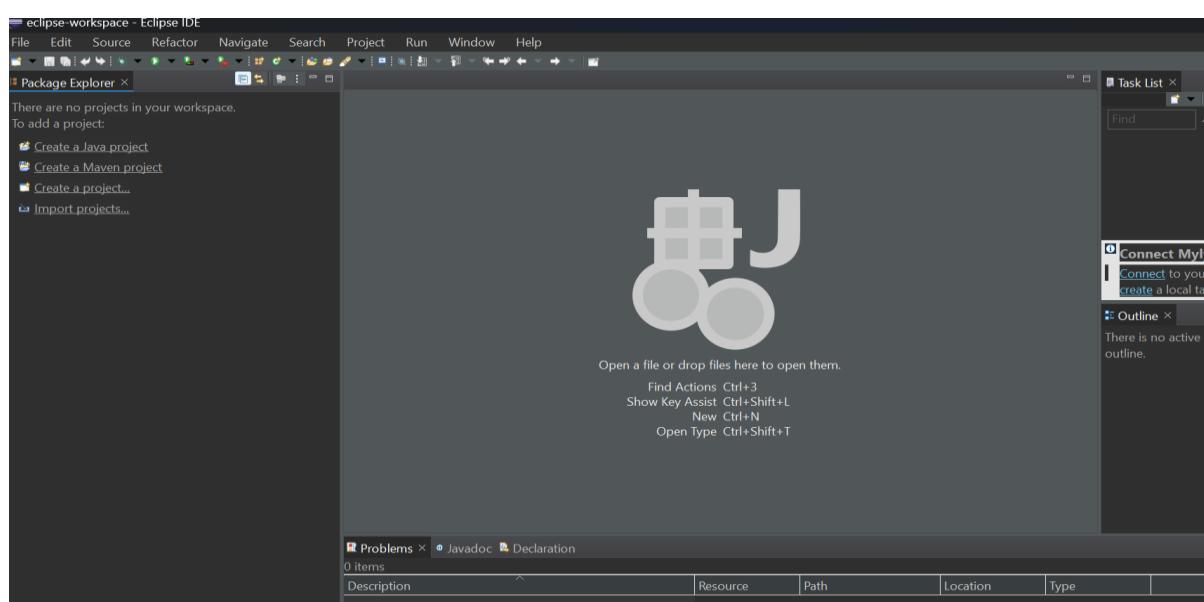
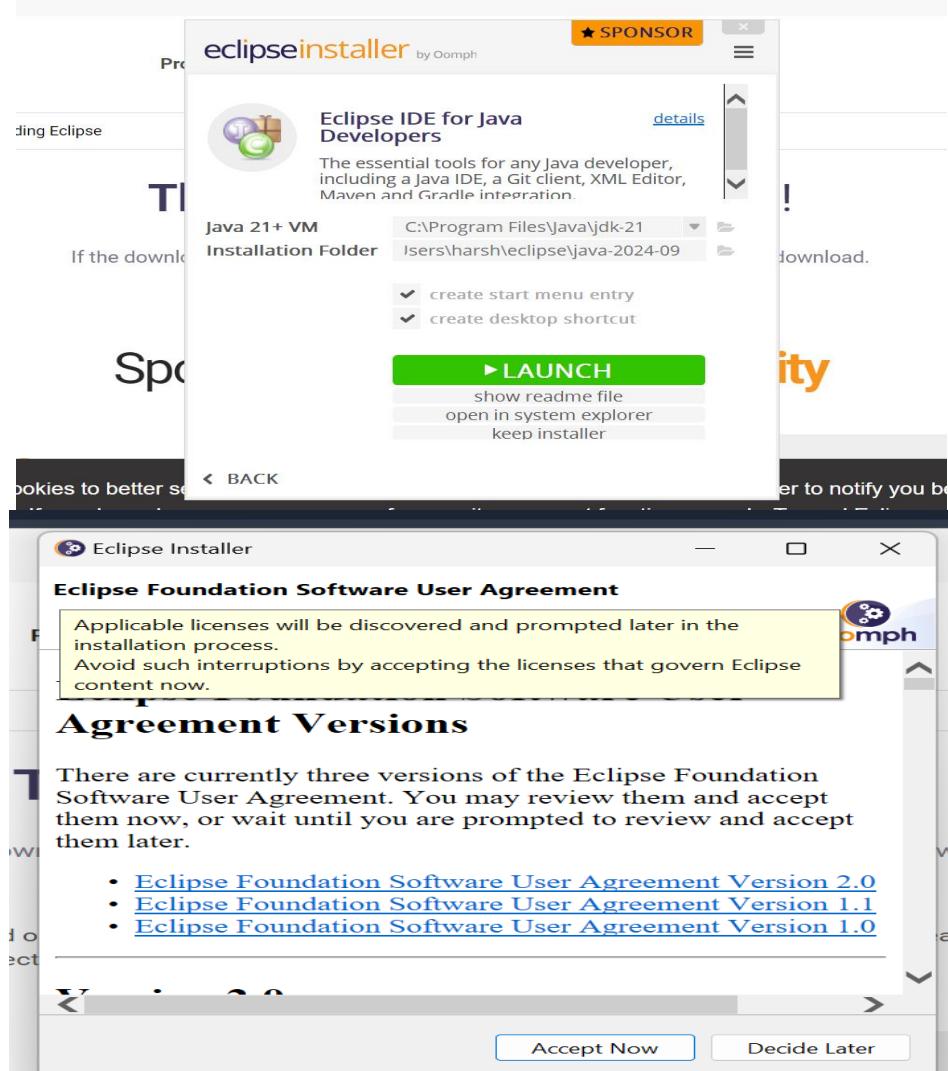


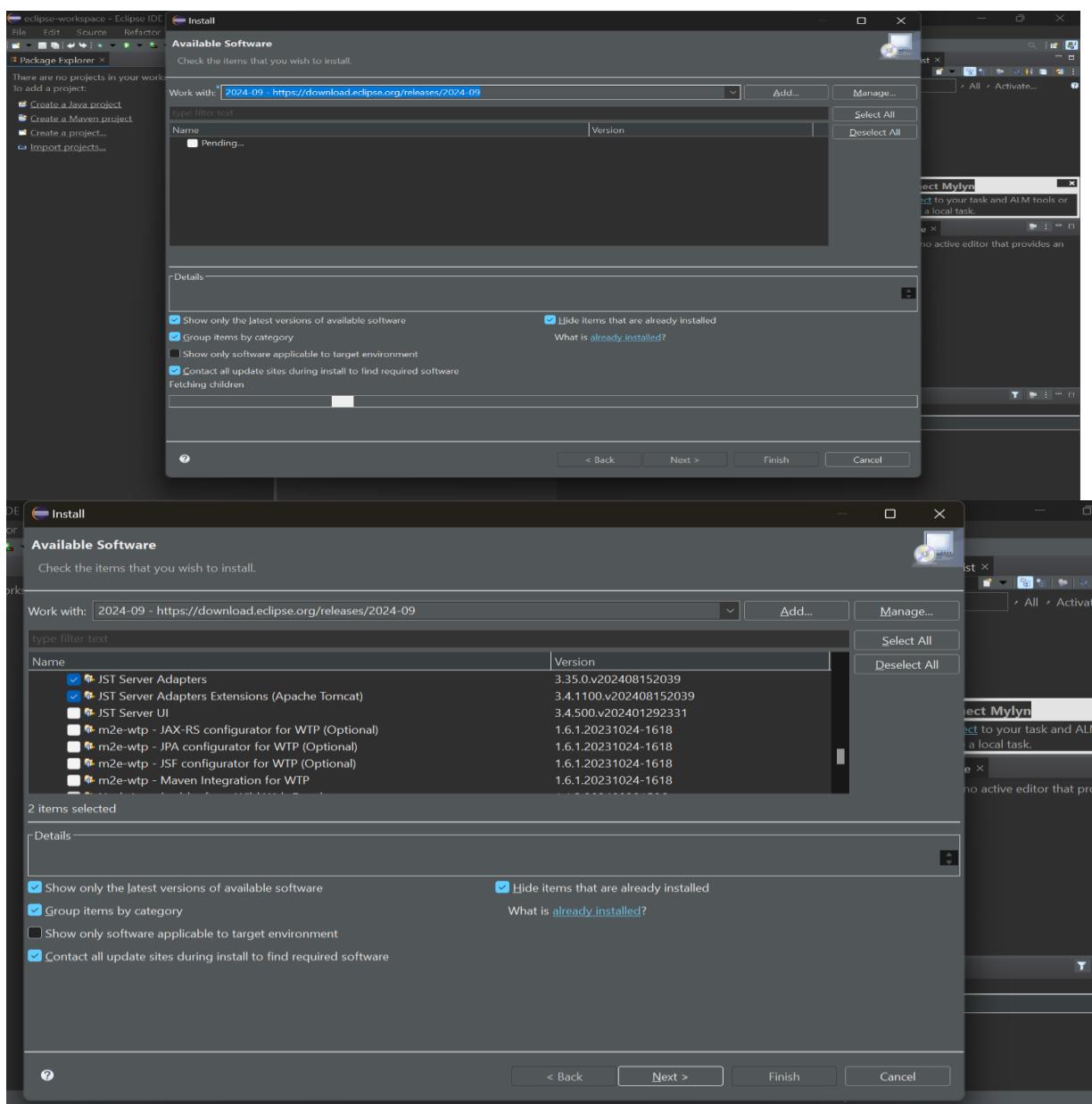
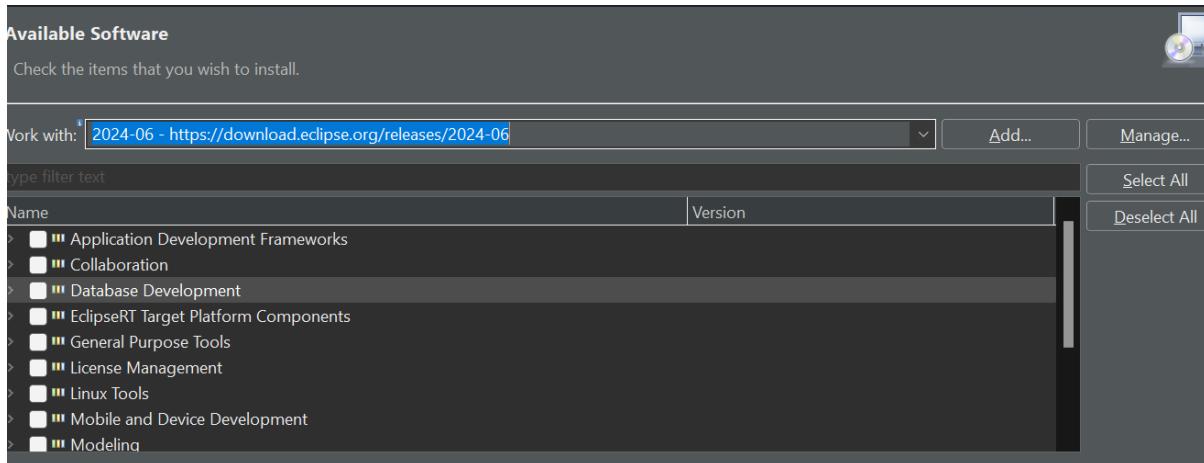
```
C:\ Command Prompt
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\laksh>mvn -version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: C:\Users\laksh\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9
Java version: 21.0.4, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "Windows"

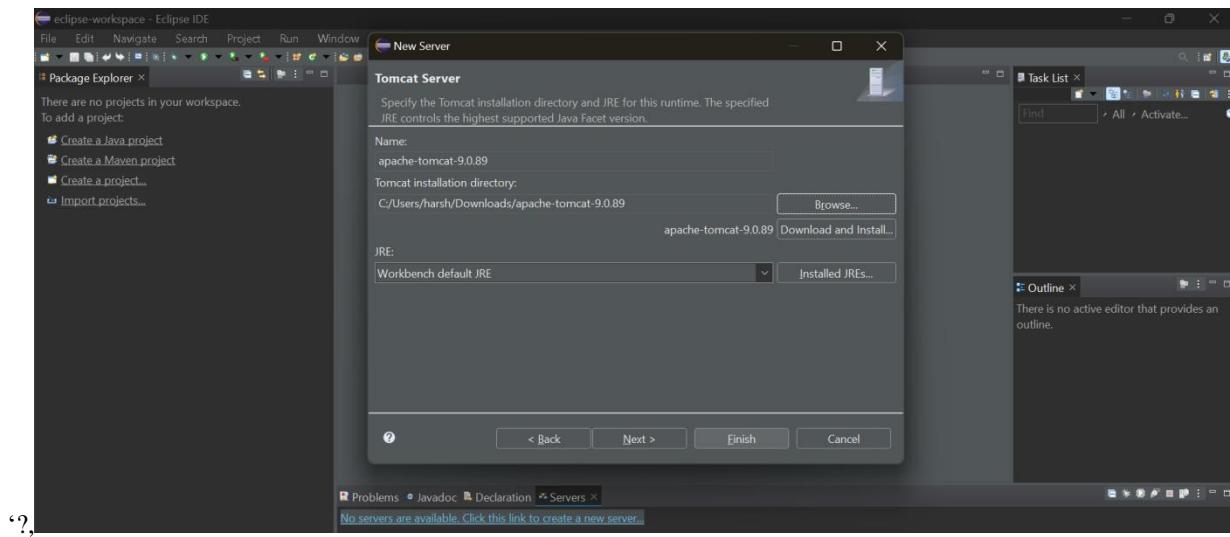
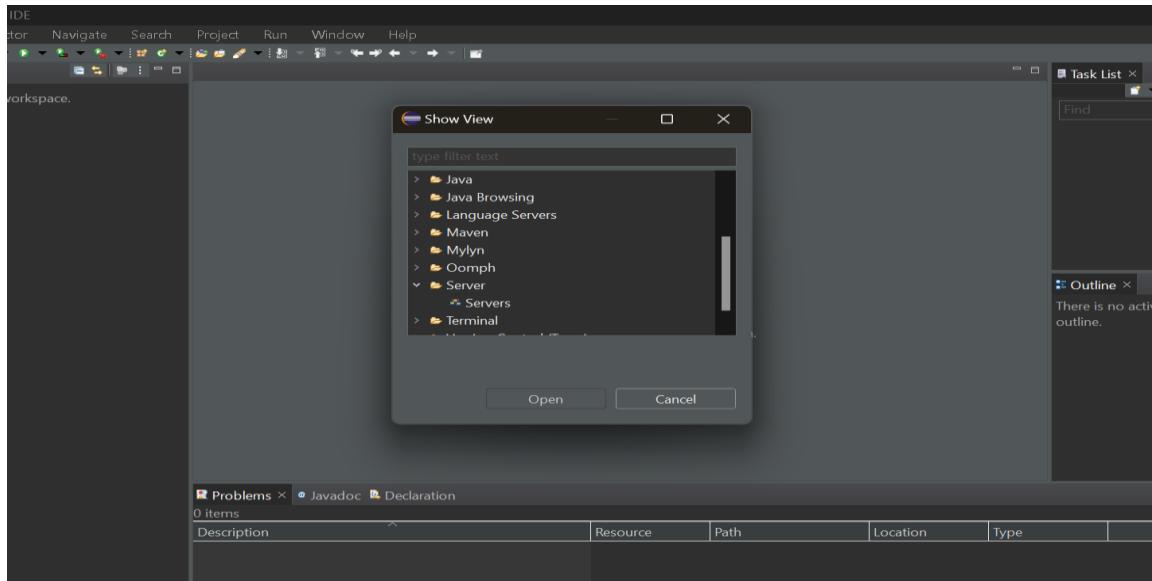
C:\Users\laksh>
```

Eclipse IDE INSTALLATION:





TOMCAT CONFIGURATION:



Experiment 4:

BASIC GIT COMMANDS - VERSION, CONFIG, INIT, STATUS, ADD, COMMIT, DIFF, HELP
GIT COMMANDS: WORKING WITH LOCAL AND REMOTE REPOSITORIES - BRANCHES,
CHECKOUT, MERGE, REVERT, LOG
GIT COMMANDS: WORKING WITH REMOTE REPOSITORIES - REMOTE, CLONE, PULL, PUSH, FORK

Git Version:

```
laksh@luxx MINGW64 ~/OneDrive/Desktop/example
$ git --version
git version 2.46.0.windows.1
laksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$
```

Git Help:

```
laksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git --help
usage: git [--version] [-h | --help] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [-no-replace-objects] [--no-lazy-fet
ch]
           [--no-optimal-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:
start a working area (see also: git help tutorial)
clone      Clone a repository into a new directory
init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
add        Add file contents to the index
mv        Move or rename a file, a directory, or a symlink
restore    Restore working tree files
rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
bisect    Use binary search to find the commit that introduced a bug
diff      Show changes between commits, commit and working tree, etc
grep      Print lines matching a pattern
log       Show commit logs
show      Show various types of objects
status    Show the working tree status

grow, mark and tweak your common history
branch   List, create, or delete branches
commit   Record changes to the repository
merge    Join two or more development histories together
rebase   Reapply commits on top of another base tip
reset   Reset current HEAD to the specified state
switch  Switch branches
tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
fetch   Download objects and refs from another repository
pull    Fetch from and integrate with another repository or a local branch
push    Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>', or 'git help <concept>'.
See 'git help git' for an overview of the system.
laksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ |
```

Git Config:

```
laksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git config --global user.name
Laksh, Vijayvargiya
laksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$
```

```

laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
Credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
core.editor="C:\Users\laksh\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=Laksh, Vijayvargiya
user.email=lakshvijay04@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

```

Creation of a file, add, status, restore, reset:

```

MINGW64:/c/Users/laksh/OneDrive/Desktop/example
laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ nano a.txt

laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ git status
On branch master
No commits yet

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

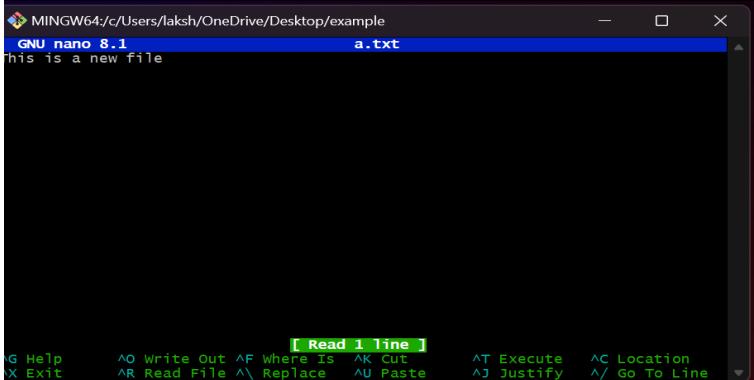
nothing added to commit but untracked files present (use "git add" to track)

laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ git add a.txt
warning: in the working copy of 'a.txt', LF will be replaced by CRLF the next time Git touches it
laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ git status
On branch master
No commits yet

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

laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ git restore a.txt
laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example (master)
$ git reset a.txt

```



List :

```

MINGW64:/c/Users/laksh/OneDrive/Desktop/example
$ ls

laksh@Ishikakakakaaaa MINGW64 ~/OneDrive/Desktop/example
$ ls -la
total 4
drwxr-xr-x 1 laksh 197609 0 Oct 15 14:28 ./
drwxr-xr-x 1 laksh 197609 0 Oct 15 14:28 ../

```

Git init and commit:

```

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git init
Reinitialized existing Git repository in C:/Users/Taksh/OneDrive/Desktop/example/.git/
Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   a.txt

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ 

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git commit -m "commit"
[master (root-commit) 1b20c34] commit
 1 file changed, 1 insertion(+)
 create mode 100644 a.txt
Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ |

```

Git branch & new file in it:

```

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git commit -m "commit"
[master (root-commit) 1b20c34] commit
 1 file changed, 1 insertion(+)
 create mode 100644 a.txt

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git branch project_changes

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git checkout project_changes
Switched to branch 'project_changes'

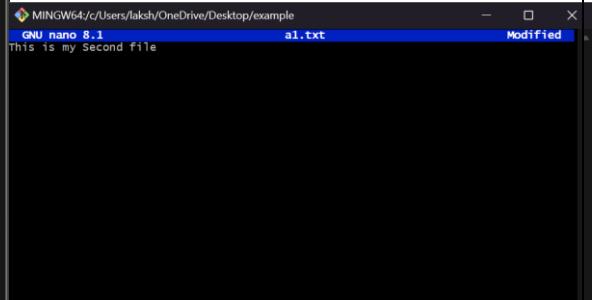
Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ nano a1.txt

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git add .
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'?
hint: Disable this message with "git config advice.addEmptyPathspec false"

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git add a1.txt
warning: in the working copy of 'a1.txt', LF will be replaced by CRLF the next time
Git touches it

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ |

```



Git log , oneline log and Git Merge:

```

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ ls
a.txt  a1.txt

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git log
commit 1b20c342494c4efa7991c83ac5776a62fe8c4a9d (HEAD -> project_changes, master)
Author: Laksh, Vijayvargiya <lakshvijay04@gmail.com>
Date:   Thu Oct 24 10:37:39 2024 +0530

    commit

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git log --oneline
1b20c34 (HEAD -> project_changes, master) commit

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ 
-

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git merge project_changes
Already up to date.

```

Git revert:

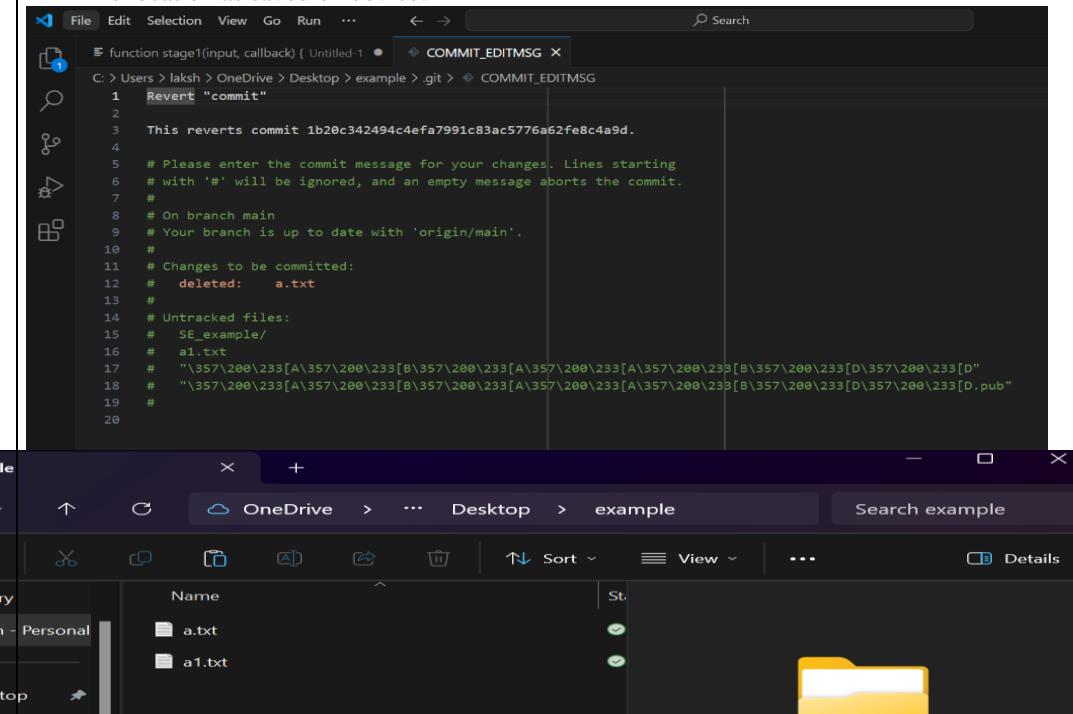
```

laksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git log --oneline
1b20c34 (HEAD -> main, origin/main, project_changes) commit

laksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git revert 1b20c34
hint: Waiting for your editor to close the file...

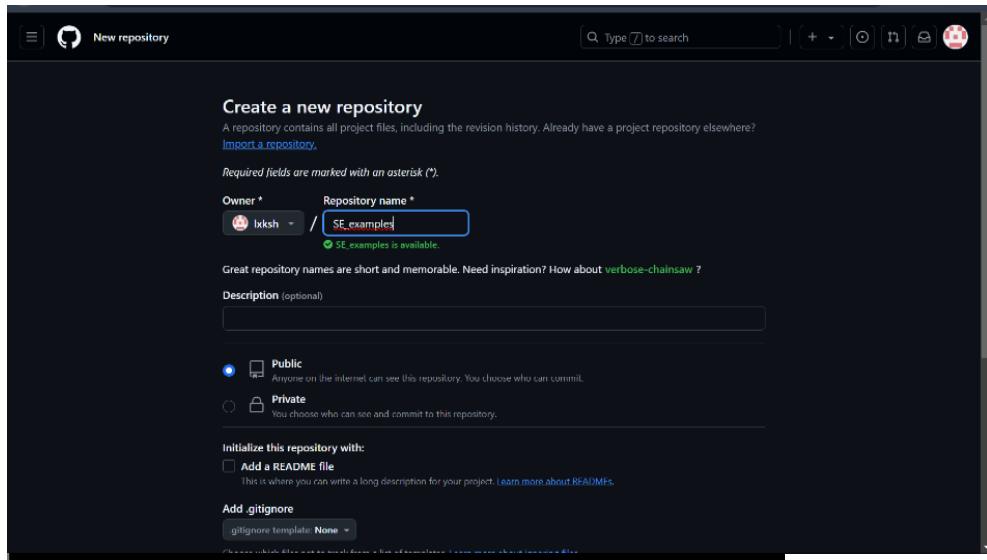
```

File location as saved on device:



Git remote:

The GitHub repository page for 'SE_example' shows various options and instructions. It includes sections for GitHub Copilot setup, adding collaborators, and quick setup instructions for cloning or creating a new repository via command line.



The screenshot shows the GitHub 'Create a new repository' interface. The repository name is 'SE_examples'. The owner is 'lxksh'. The repository is set to 'Public'. A README file is added. The terminal below shows the steps to initialize a local repository, add it to GitHub, and push it to the remote origin.

```
taksh@luxx MINGW64 ~/OneDrive/Desktop/example (project_changes)
$ git checkout master
Switched to branch 'master'
A     a1.txt

taksh@luxx MINGW64 ~/OneDrive/Desktop/example (master)
$ git branch -M main

taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git remote add origin https://github.com/lxksh/SE_example.git

taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git remote -v
origin  https://github.com/lxksh/SE_example.git (fetch)
origin  https://github.com/lxksh/SE_example.git (push)

taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ |

taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git push -u origin main

Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 232.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lxksh/SE_example.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$
```

SSH key generation:

```
Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ ssh-keygen -t rsa -C lakshvijay04@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/users/laksh/.ssh/id_rsa):
Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ ssh-keygen -t rsa -C lakshvijay04@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/users/laksh/.ssh/id_rsa): /c/Users/laksh/.ss
h/id_rsa
Created directory '/c/users/laksh/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/users/laksh/.ssh/id_rsa
Your public key has been saved in /c/users/laksh/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:rJi8JNq+vH+qucfLAAoJIFFYjBBa14SM7HgdPjBqMhA lakshvijay04@gmail.com
The key's randomart image is:
+--[RSA 3072]--+
|E+.o+.
|oo=+.
|o+o=+.
|=o+ .
|=o+ . S
|++ + o .
|+ ..* .
|+ =oo.
|..XB*+
+---[SHA256]---+
Taksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$
```

ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDTMrxwvJhjGQ5ic+
32Daln52pXPWqrwPHRci1iuTwYOpje9GM0nLr38BDa1gTxIXtNwXCF+w1Gm2pD00L/10LGtWu3
+MVXShc3fWcacMnsCtWasCvUD+yET9u744YmNt5XPRP6IjHmo802gBj9nDhhURxkT1ae3sI1q26hdFu66yzfx3aHJCJyboUUTMJKgRFfwFQtHyuhBD/
VVvgjY6eZgefmpVgtiSh1htTcUV4VzQnyWA89tB/Rnj0gtl39tnU8YQx9gpVr1J3/Xigig4QY1L4VHQqrS+sYo9cBN0c41v1+rwxFi0uOBOLwHmrPiELO
56ckUKc0m5GsqzxBVeaWA/JUiKtufAGi/hA+
9pSo2ab512Dpj3gyAE09PPVsgg/mto+dQ3zkH7PEI5QFEpbp4ooD6XITYHEfg7gjPpkf50wJQYqf91V++C7VxIzGeAwsrWk7EY3RH8G1cEMyn9z6yAMC
KD1KGgridHmq4DIJFmXmrpxg6VXW/juvaxQrbXc= lakshvijay04@gmail.com

Name	Date
id_rsa	24
id_rsa.pub	24

The screenshot shows the GitHub 'Settings' page under the 'SSH and GPG keys' section. A modal window titled 'Add new SSH Key' is open, prompting for a 'Title' and 'Key type' (set to 'Authentication Key'). The 'Key' field contains a long string of characters representing an SSH key. At the bottom of the modal is a green 'Add SSH key' button.

You have successfully added the key 'lakshvijay04@gmail.com'.

The screenshot shows the 'SSH keys' section of the GitHub settings. It lists one key associated with the email 'lakshvijay04@gmail.com'. The key has a SHA-256 fingerprint, was added on Oct 24, 2024, and is marked as 'Never used — Read/write'. A 'Delete' button is visible next to the key entry.

Git Clone:

The screenshot shows a GitHub repository named 'SE_example' with a single commit containing a file 'a.txt'. A context menu is open over the commit, with the 'Clone' option selected. The URL 'git@github.com:laksh/SE_example.git' is shown. Below the repository is a file explorer window showing a folder named 'example' containing files 'a.txt', 'a1.txt', and two partially visible files starting with '·[A·[B·[A·[A·[B·[D·[D·[D.pub'.

```

laksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ git clone git@github.com:lxksh/SE_example.git
Cloning into 'SE_example'...
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+diy3wvvv6TuJhbpZisF/zLDAOzPMSVHdkr4UvCoQU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
Receiving objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)

laksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ ls -ltr
total 14
-rw-r--r-- 1 laksh 197609 20 Oct 15 14:33 a.txt
-rw-r--r-- 1 laksh 197609 23 Oct 24 10:39 a1.txt
-rw-r--r-- 1 laksh 197609 564 Oct 24 12:14 ''$'\033''[A'$'\033''[B'$'\033''[A'$'\03
3''[A'$'\033''[B'$'\033''[D'$'\033''[D.pub'
-rw-r--r-- 1 laksh 197609 2590 Oct 24 12:14 ''$'\033''[A'$'\033''[B'$'\033''[A'$'\03
3''[A'$'\033''[B'$'\033''[D'$'\033''[D'
drwxr-xr-x 1 laksh 197609 0 Oct 24 12:25 SE_example/
laksh@luxx MINGW64 ~/OneDrive/Desktop/example (main)
$ cd SE_example
laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ 

laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ ls -ltr
total 1
-rw-r--r-- 1 laksh 197609 21 Oct 24 12:25 a.txt
laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ |

```

Git Push:

```

laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git add .

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git commit -m "Added a new server file"
[main 7da308e] Added a new server file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 server.txt

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git remote -v
origin  git@github.com:lxksh/SE_example.git (fetch)
origin  git@github.com:lxksh/SE_example.git (push)

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:lxksh/SE_example.git
  1b20c34..7da308e  main -> main

Taksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ 

```

The terminal window shows the command `git pull` being run, indicating a successful update from the remote repository. The GitHub repository page for 'SE_example' shows two commits: 'a.txt' and 'server.txt'. A 'README' section is present with a 'Add a README' button.

Git Pull:

```

laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git remote -v
origin  git@github.com:lxksh/SE_example.git (fetch)
origin  git@github.com:lxksh/SE_example.git (push)

laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ ls -ltr
total 1
-rw-r--r-- 1 laksh 197609 21 Oct 24 12:25 a.txt
-rw-r--r-- 1 laksh 197609 0 Oct 24 12:29 server.txt

laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ git pull origin main
From github.com:lxksh/SE_example
 * branch      main            -> FETCH_HEAD
Already up to date.

laksh@luxx MINGW64 ~/OneDrive/Desktop/example/SE_example (main)
$ |
```

The file explorer shows the contents of the 'SE_example' folder, which contains 'a.txt' and 'server.txt' files. Both files were modified on 24-Oct-2024 at 12:25 and 12:29 respectively.

Name	Status	Date modified	Type
a.txt	●	24-10-2024 12:25	Text Document
server.txt	●	24-10-2024 12:29	Text Document

Create a fork:

Screenshot of GitHub showing the archiving of the `orf/simple` repository and its fork by `lksh`.

Top Panel (Archived Repository):

- Repository: `orf/simple`
- Owner: `orf`
- Archived by: `orf` on Oct 26, 2018
- Commits: 278
- Branches: 4
- Tags: 0
- Issues: 4
- Pull requests: 2
- Actions: 0
- Projects: 0
- Wiki: 0
- Security: 0
- Insights: 0

Middle Panel (Create a new fork):

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner: `lksh` / **Repository name:** `simple` (available)

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional): Simple is a clone of Obtuse written in Python running on Flask.

Copy the master branch only: Contribute back to `orf/simple` by adding your own branch. [Learn more.](#)

ⓘ You are creating a fork in your personal account.

Create fork

Bottom Panel (Forked Repository):

Repository: `lksh/simple`

Forked from: `orf/simple`

Code | **Pull requests** | **Actions** | **Projects** | **Wiki** | **Security** | **Insights** | **Settings**

Simple (Public)

forked from [orf/simple](#)

Branches: master | 1 Branch | Tags

This branch is up to date with `orf/simple:master`

Commits:

Author	Commit Message	Date	Commits
<code>orf</code>	Merge pull request <code>orf#54</code> from <code>adamchainz/patch-1</code>	6 years ago	278
<code>simple</code>	Change url protocol for more secure access	8 years ago	
<code>.gitignore</code>	Bump version and fix nginx_config from not working	10 years ago	
<code>LICENSE</code>	Add MIT license	10 years ago	
<code>MANIFEST.in</code>	Add Manifest.in	10 years ago	
<code>README.md</code>	Update README for maintenance status	6 years ago	
<code>setup.py</code>	Change url protocol to https for secure access	8 years ago	

Experiment 5:

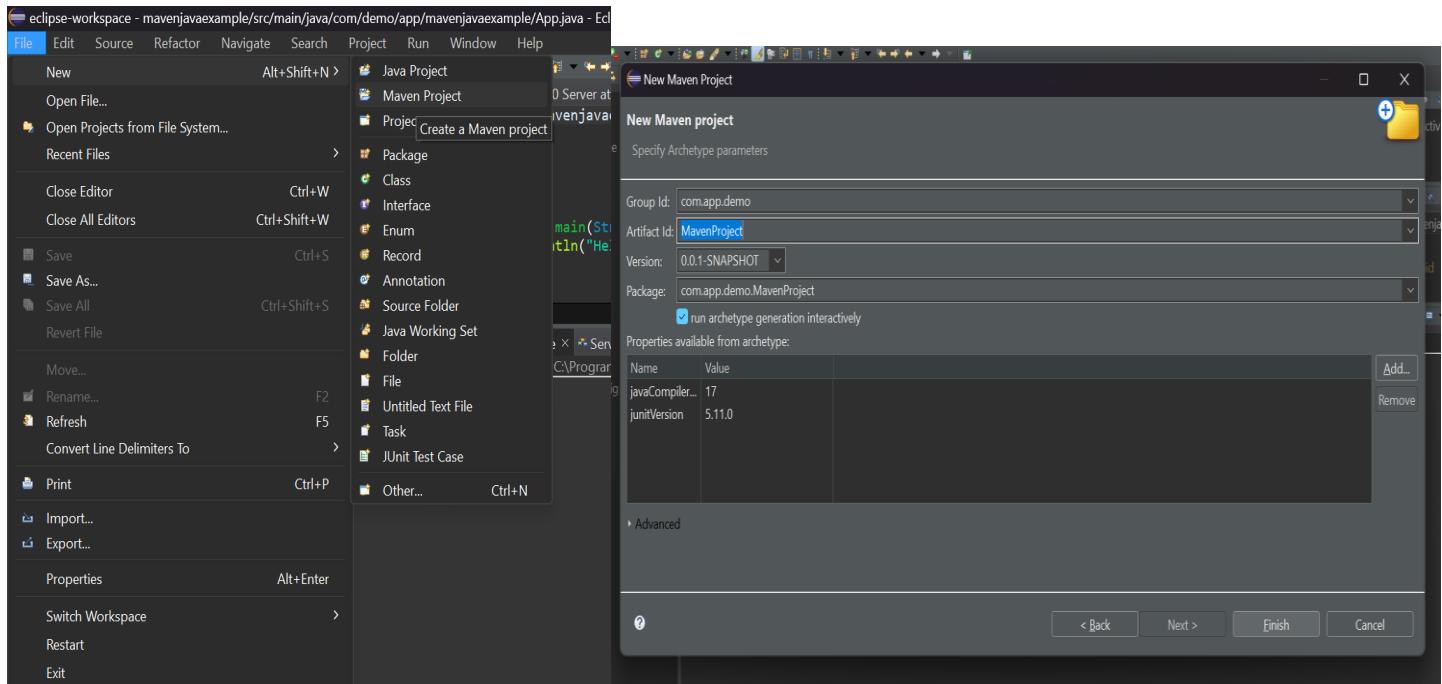
CREATING MAVEN JAVA PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB.

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

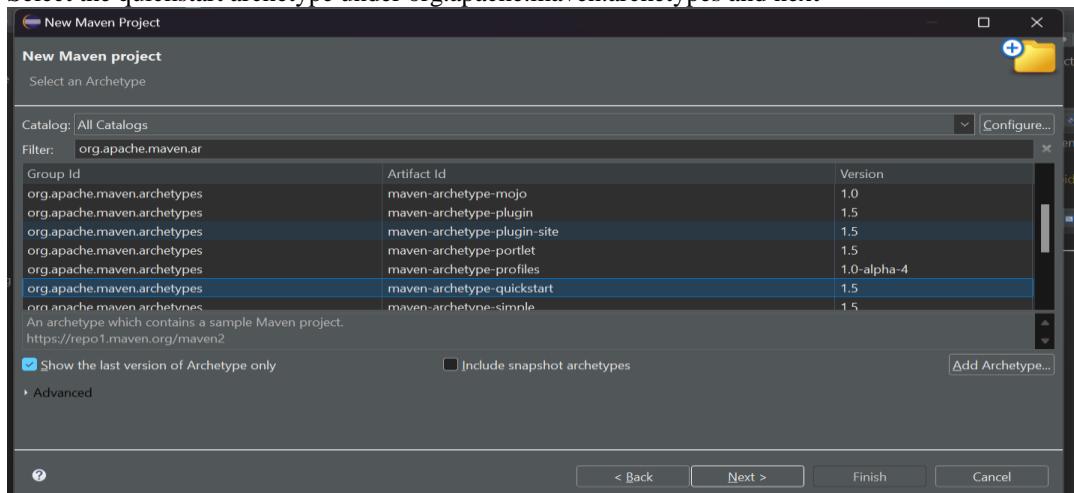
INSTALLATION OF JENKINS.

1. MAVEN JAVA PROJECT

Create a Maven project -> File -> New -> Maven -> Maven Project: Click on Next



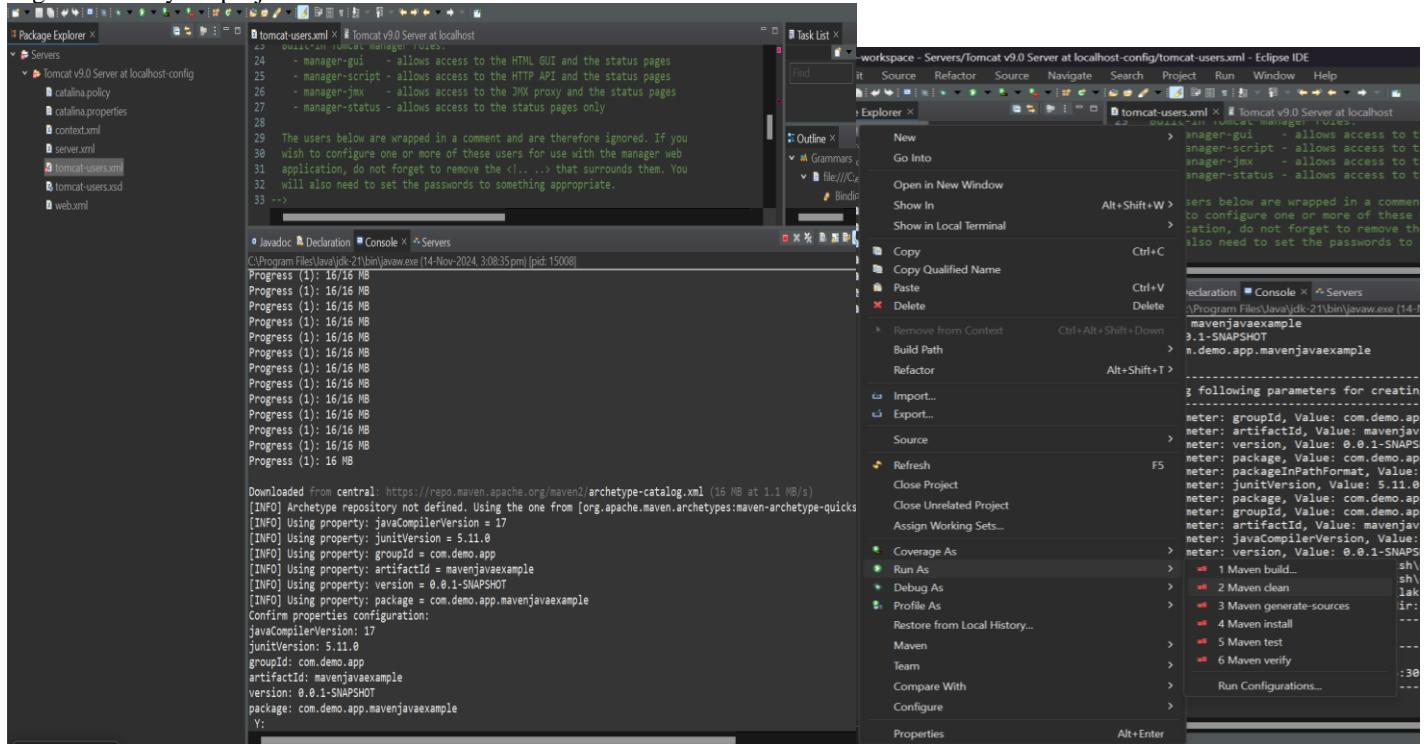
Select the quickstart archetype under org.apache.maven.archetypes and next



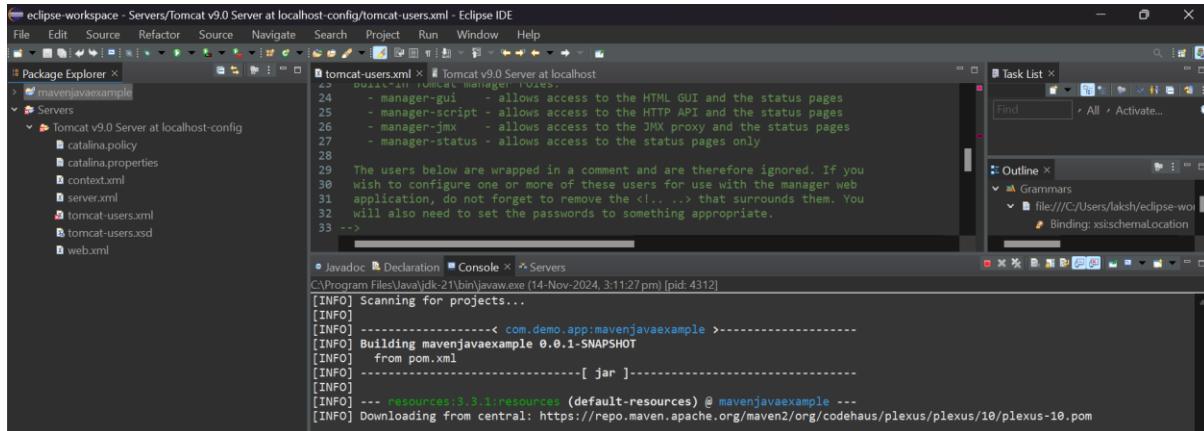
Type Y to create a project and check for App.java



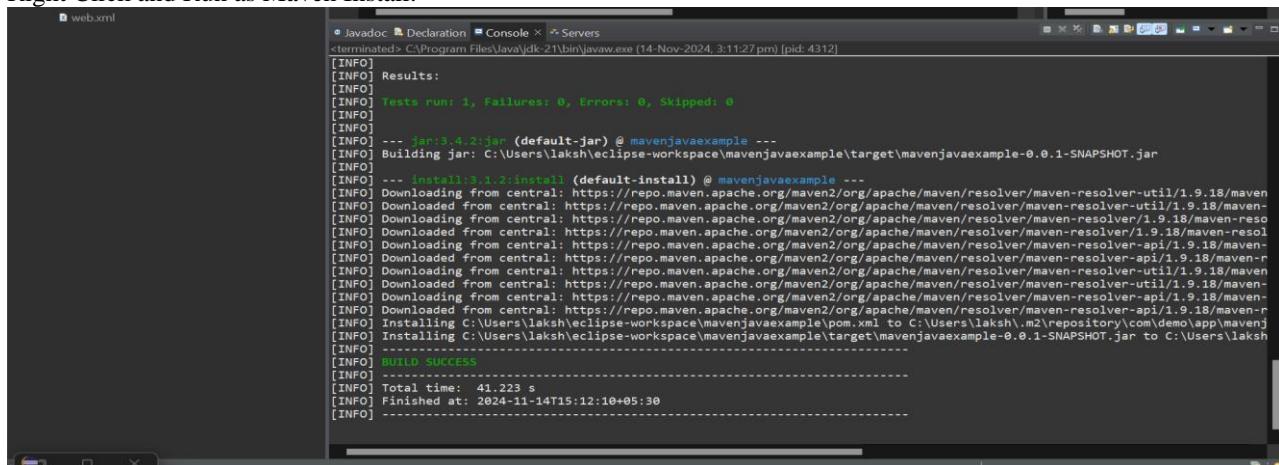
Right click on your project to run as maven clean



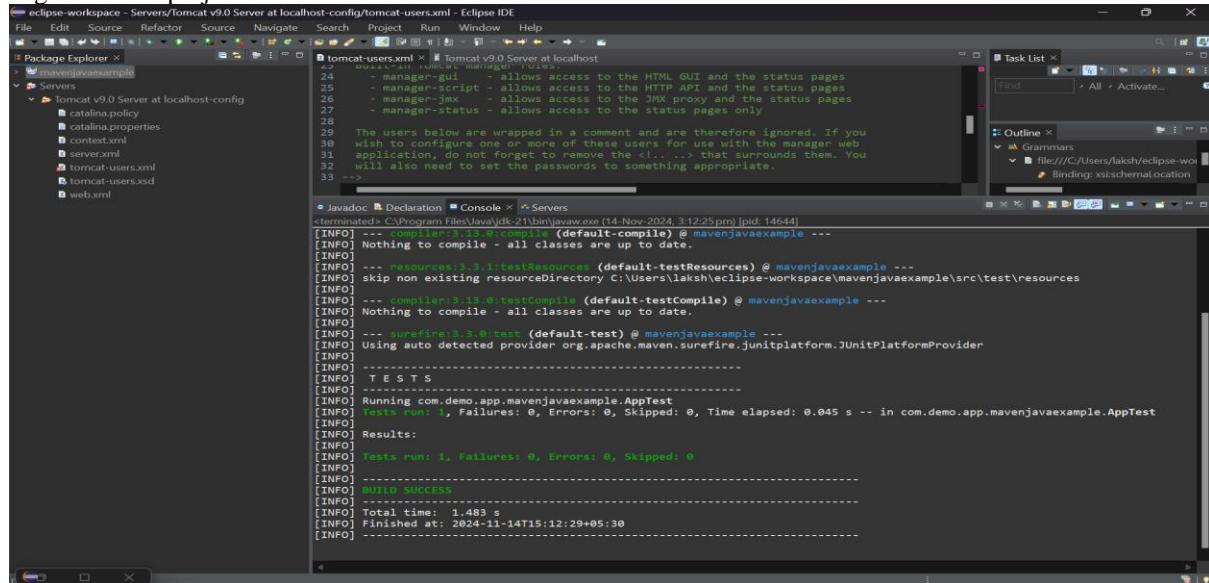
Check the console for Maven Build Success:



Right Click and Run as Maven Install:



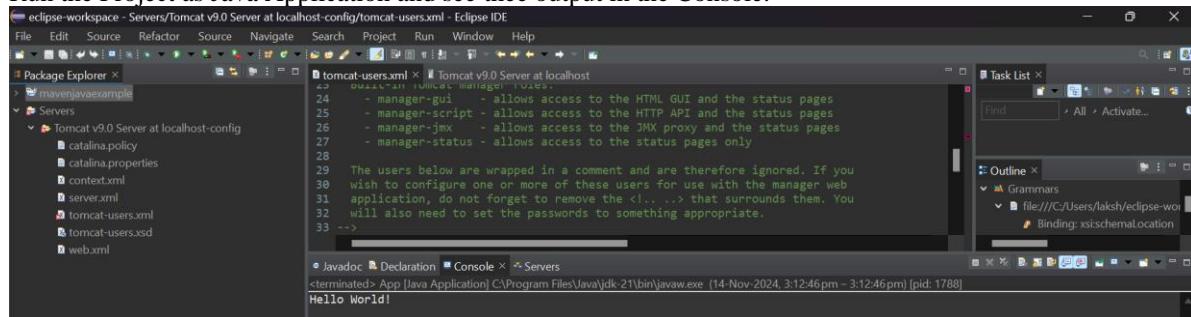
Right click on project and click on Run as Maven Test:



The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view on the left displaying a 'mavenjavaexample' project under 'Servers'. The central area shows the 'Console' tab with the following Maven test output:

```
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ mavenjavaexample ---
[INFO] skip non existing resourceDirectory C:\Users\laksh\workspace\mavenjavaexample\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mavenjavaexample ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.0.0-test (default-test) @ mavenjavaexample
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----
[INFO] Running com.demo.app.mavenjavaexample.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] Results:
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.483 s
[INFO] Finished at: 2024-11-14T15:12:29+05:30
[INFO]
```

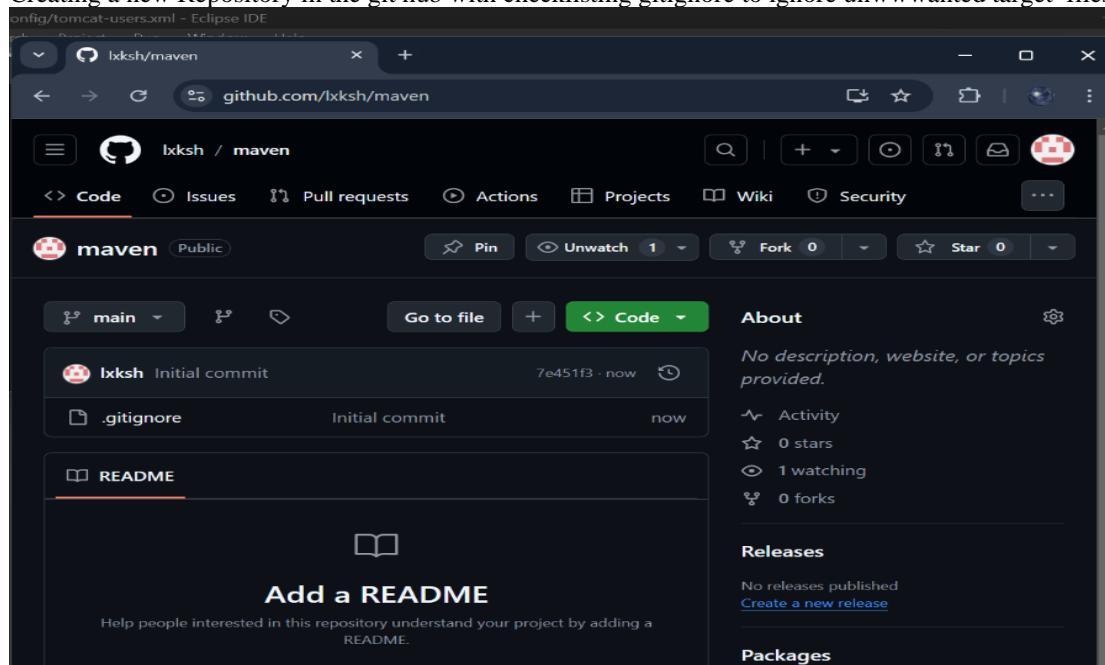
Run the Project as Java Application and see thee output in the Console:



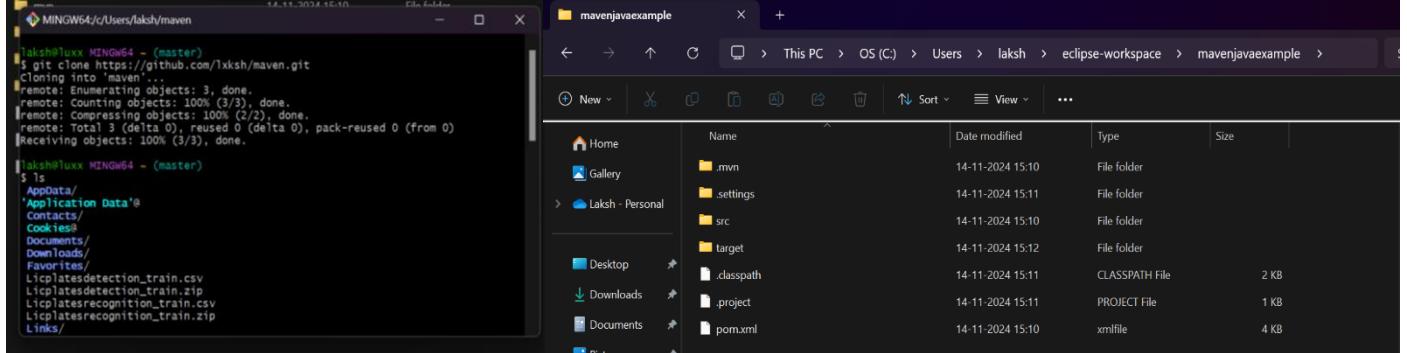
The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view on the left displaying a 'mavenjavaexample' project under 'Servers'. The central area shows the 'Console' tab with the following Java application output:

```
terminated: App [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (14-Nov-2024, 3:12:46 pm - 3:12:46 pm) [pid: 1788]
Hello World!
```

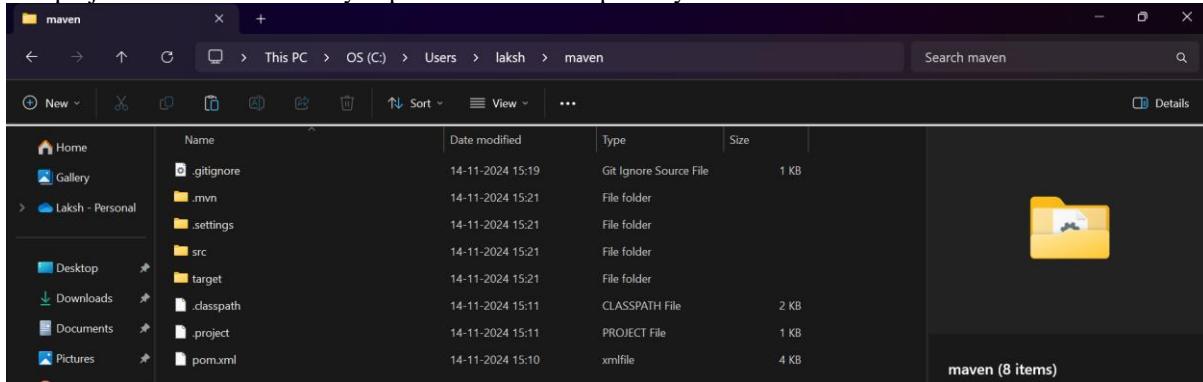
Creating a new Repository in the git hub with checklistng gitignore to ignore unwwwanted target files and it



Clone the GitHub repository into the local System: Cloning local & global Projects to copy from eclipse to local cloned Repository:



The project files are successfully copied into Maven Repository



Using Gitbash add and commit the changes Made:

```

MINGW64:/c/Users/laksh/maven
licenseplate.ipynb
maven/
notebook3b4d8ad441.ipynb
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
test.zip
untitled.txt

Taksh@lxx MINGW64 ~ (master)
$ cd maven

Taksh@lxx MINGW64 ~/maven (main)
$ git add .

Taksh@lxx MINGW64 ~/maven (main)
$ git commit -m "First Commit"
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean

Taksh@lxx MINGW64 ~/maven (main)
$
```

Push the Changes into the Github as shown

```

Taksh@lxx MINGW64 ~/maven (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .mvn/
    .settings/
    pom.xml/
    src/

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

Taksh@lxx MINGW64 ~/maven (main)
$ git add .

Taksh@lxx MINGW64 ~/maven (main)
$ git commit -m "First Commits"
[main 2f7c720] First Commits
 8 files changed, 353 insertions(+)
  create mode 100644 .mvn/maven.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 pom.xml
  create mode 100644 src/main/java/com/demo/app/mavenjavalexample/App.java
  create mode 100644 src/test/java/com/demo/app/mavenjavalexample/AppTest.java

Taksh@lxx MINGW64 ~/maven (main)
$ git push
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 16 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (24/24), 2.61 KiB | 890.00 KiB/s, done.
Total 24 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lkxsh/maven.git
 7e451f3..2f7c720 main -> main

```

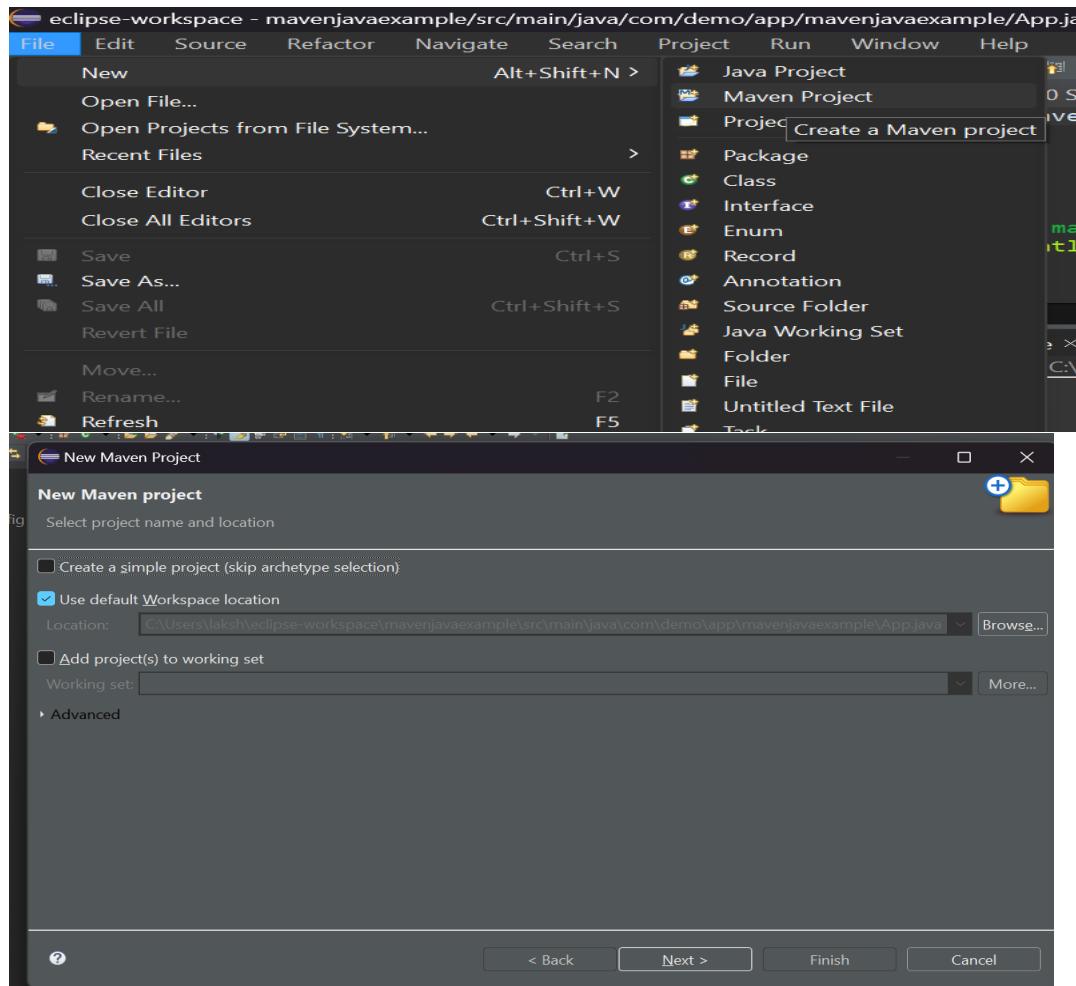
Refresh the GitHub to see the updated repository:

The GitHub repository page shows the following details:

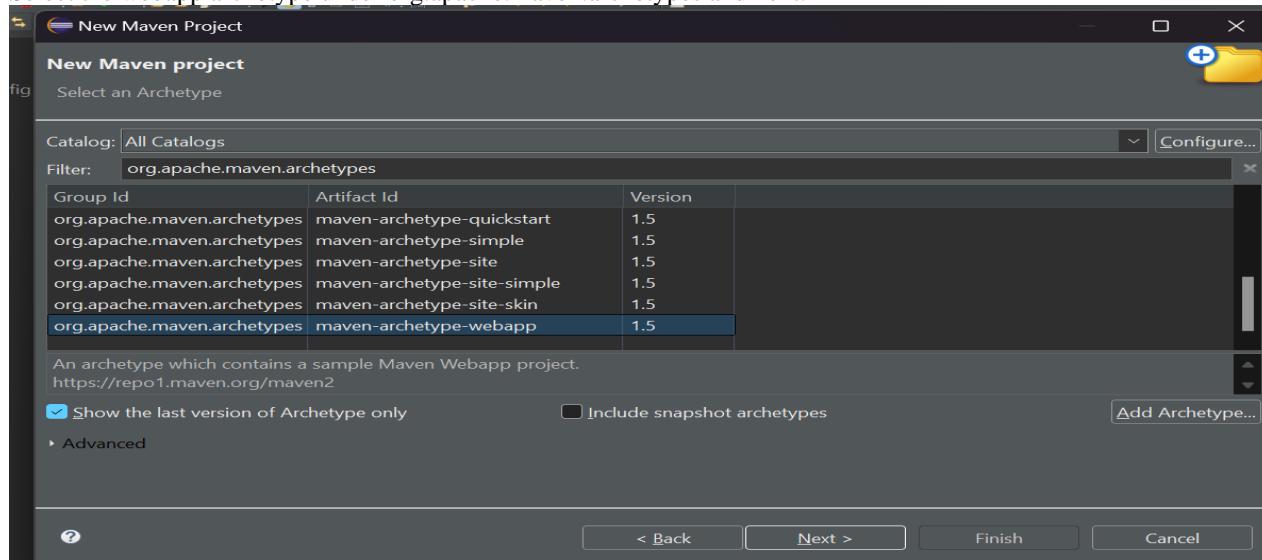
- Code**, **Issues**, **Pull requests**, **Actions**, **Projects**, **Wiki**, **Security**, **Insights**, **Settings**
- maven** (Public)
- main** Branch: 1 Branch, 0 Tags
- lkxsh** First Commits (Commit ID: 2f7c720, now)
- Initial commit** (17 minutes ago)
- README**

2. WEB PROJECT

Create a Maven project -> File -> New -> Maven -> Maven Project: Click on Next:



Select the webapp archetype under org.apache.maven.archetypes and next:



WebProject is Successfully build:

```

Javadoc Declaration Console × Servers
<terminated> C:\Program Files\Java\jdk-21\bin\javaw.exe (14-Nov-2024, 4:04:46pm) [pid: 20572]
Progress (1): 16/16 MB
Progress (1): 16 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (16 MB at 2.3 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.5] found in ca
[INFO] Using property: groupId = com.app.demo
[INFO] Using property: artifactId = WebProject
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = com.app.demo.WebProject
Confirm properties configuration:
groupId: com.app.demo
artifactId: WebProject
version: 0.0.1-SNAPSHOT
package: com.app.demo.WebProject
Y: y
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-webapp:1.5
[INFO] -----
[INFO] Parameter: groupId, Value: com.app.demo
[INFO] Parameter: artifactId, Value: WebProject
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: com.app.demo.WebProject
[INFO] Parameter: packageInPathFormat, Value: com/app/demo/WebProject
[INFO] Parameter: package, Value: com.app.demo.WebProject
[INFO] Parameter: groupId, Value: com.app.demo
[INFO] Parameter: artifactId, Value: WebProject
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[WARNING] CP: Don't override file C:\Users\laksh\workspace\WebProject\src\main\webapp
[WARNING] CP: Don't override file C:\Users\laksh\workspace\WebProject\.mvn
[INFO] Project created from Archetype in dir: C:\Users\laksh\workspace\WebProject
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.558 s
[INFO] Finished at: 2024-11-14T16:05:03+05:30
[INFO] -----

```

Go to mvnrepository for java servlet:

Version	Vulnerabilities	Repository	Usages	Date
3.0.x 3.0-alpha-1		Central	234	Apr 17, 2008
2.5.x 2.5		Central	8,623	Jul 17, 2006
	2.4	Central	4,128	Nov 08, 2005
2.4.x 2.4.public_draft		Central	31	Nov 08, 2005
	2.4-20040521	Central	7	Nov 08, 2005
2.3.x 2.3		Central	1,063	Nov 08, 2005
2.2.x 2.2		Central	54	Nov 08, 2005

Add the dependency to the pom.xml file:

```

<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>2.5</version>
</dependency>

```

Start the server :

The screenshot shows the Eclipse IDE interface with the Maven Dependency Hierarchy view open. The dependency tree for the project 'WebProject' is displayed, showing various dependencies like JUnit, Servlet API, and Apache Commons Lang. To the right, the 'Servers' view shows the Tomcat v9.0 Server at localhost configuration. The server logs indicate the deployment of the application and the start of the Catalina container.

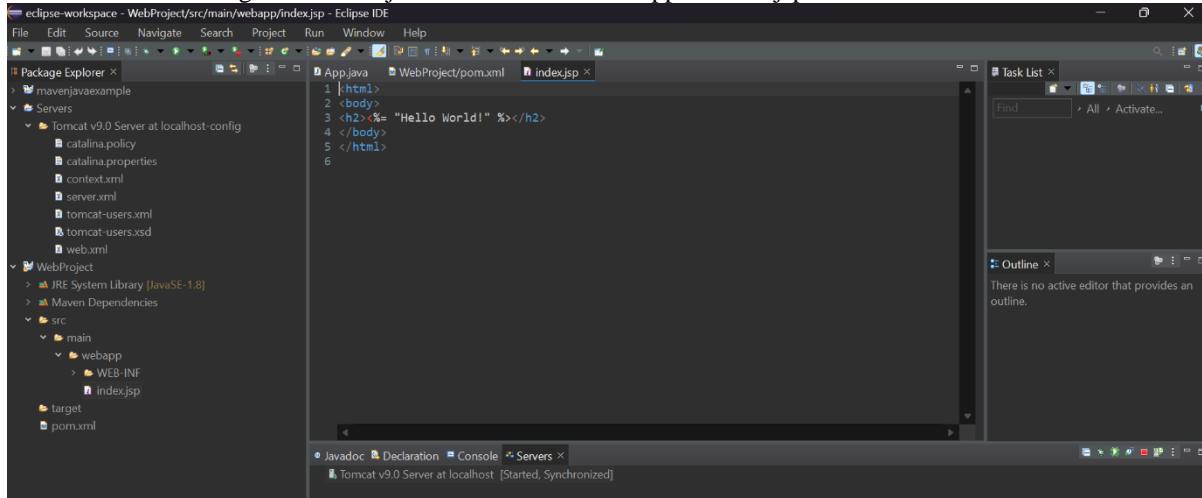
```

<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13.1</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax-servlet-api</artifactId>
    <version>2.5</version>
</dependency>
<build>
    <finalName>WebProject</finalName>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be overridden in child POMs) -->
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.6.3</version>
                <configuration>
                    <warSourceDirectory>src/main/webapp</warSourceDirectory>
                    <warOutputDirectory>target</warOutputDirectory>
                </configuration>
            </plugin>
        </plugins>
    </pluginManagement>
</build>

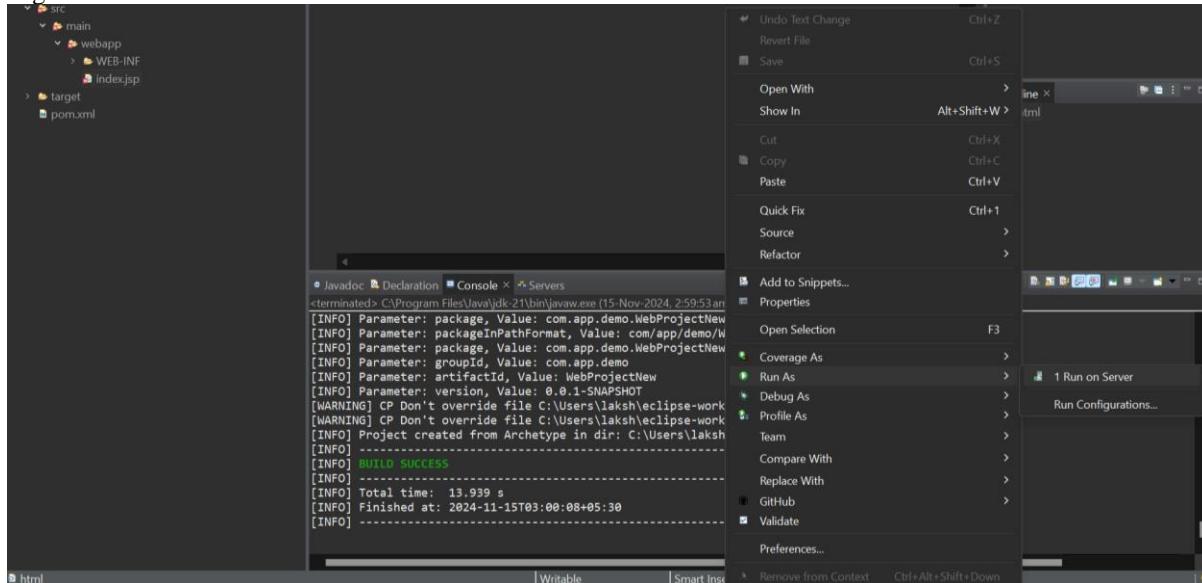
```

INFO: Deploying web application directory [C:\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager]
Nov 14, 2024 4:07:01 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory [C:\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\host-manager] has finished in 1 ms
Nov 14, 2024 4:07:01 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory [C:\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\manager]
Nov 14, 2024 4:07:01 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory [C:\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\manager] has finished in 1 ms
Nov 14, 2024 4:07:01 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory [C:\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\ROOT]
Nov 14, 2024 4:07:01 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory [C:\apache-tomcat-9.0.97-windows-x64\apache-tomcat-9.0.97\webapps\ROOT] has finished in 1 ms
Nov 14, 2024 4:07:01 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8083"]
Nov 14, 2024 4:07:01 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [712] milliseconds

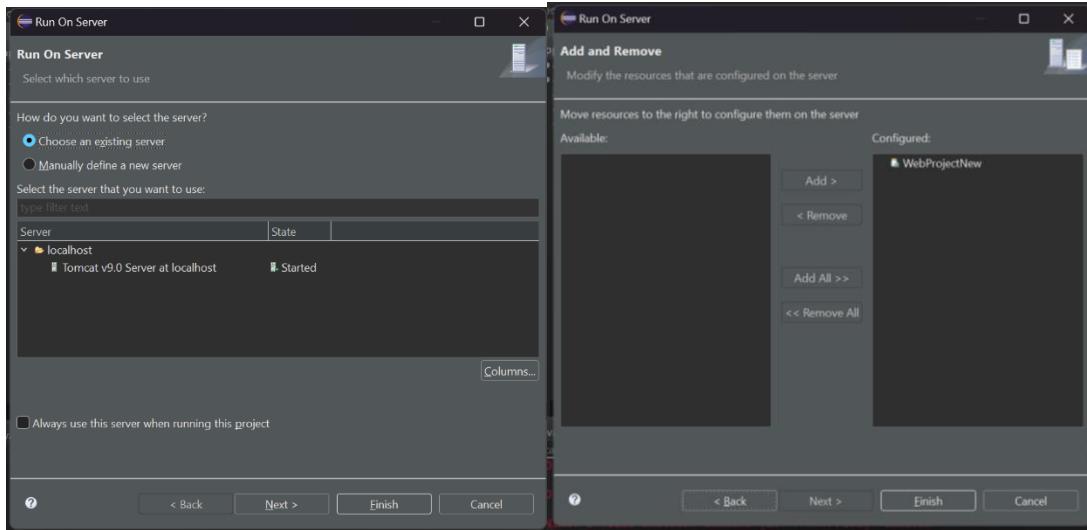
Once the server is on go to WebProject-> src-> main->webapp-> index.jsp:



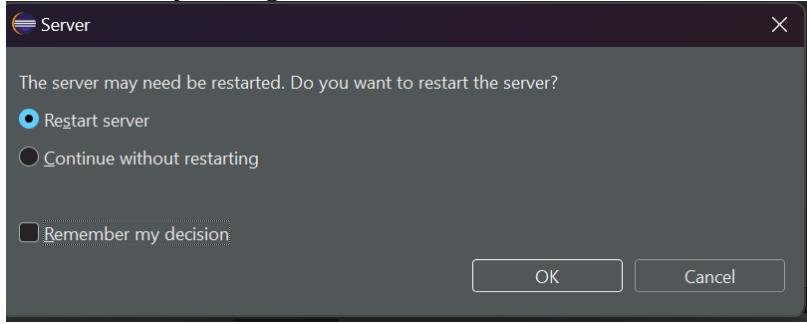
Right click on it and inn menu click on Run as Server:



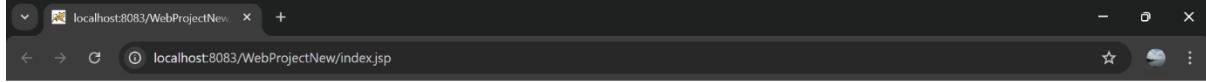
Select the Tomcat v9.0 server and click on next:



Restart server by clicking on OK:



Web Server is running and the prompt is shown:



Open tomcat manager to see running sessions:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/WebProjectNew	None specified	Archetype Created Web Application	true	2	Start Stop Reload Undeploy
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy

Create a new Repository and add a readme file in it:

The screenshot shows the GitHub 'New repository' creation interface. The repository name is 'WebProject'. The visibility is set to 'Public'. A README file is selected for initialization. Other options like 'Add a README file' and '.gitignore' are also visible.

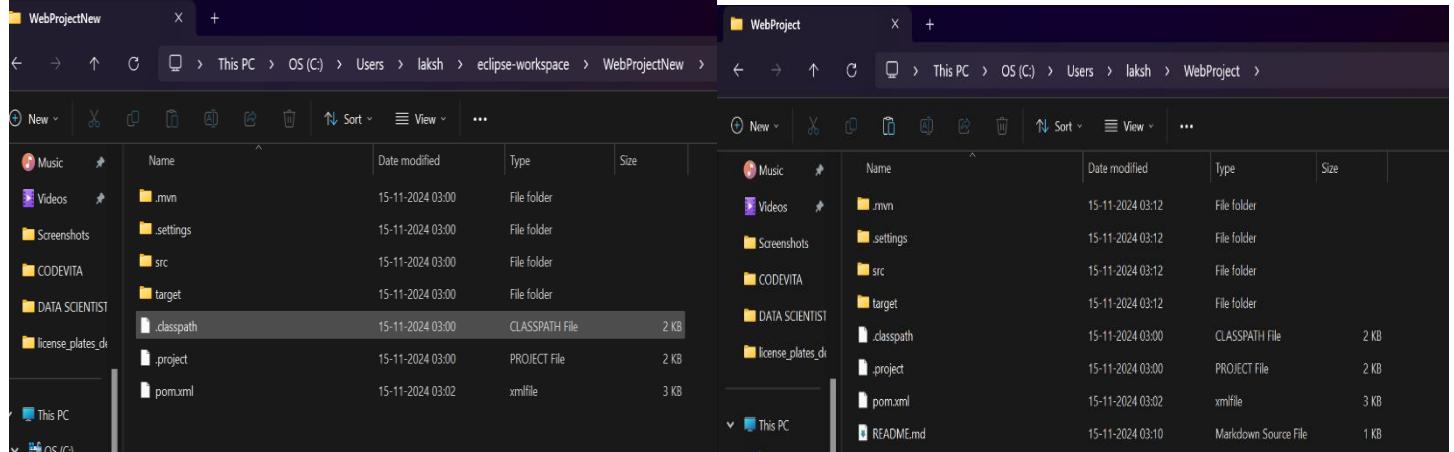
Repository is created with readme file:

The screenshot shows the GitHub repository page for 'WebProject'. It displays one commit from 'lksh' titled 'Initial commit' and a README file.

Open git bash and clone the repository into your local System:

```
Taksh@luxx MINGW64 ~ (master)
$ git clone https://github.com/lxksh/WebProject.git
```

Cloning local and global Projects to copy contents from eclipse workspace to local cloned Repository
The project files are successfully copied to WebProject repository



Open git bash and add the repository:

```
MINGW64:/c/Users/laksh/WebProject
laksh@luxx MINGW64 ~ (master)
$ git clone https://github.com/laksh/WebProject.git
Cloning into 'WebProject'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

laksh@luxx MINGW64 ~ (master)
$ cd WebProject

laksh@luxx MINGW64 ~/WebProject (main)
$ git add .
warning: in the working copy of 'src/main/webapp/WEB-INF/web.xml', LF will be
placed by CRLF the next time Git touches it
warning: in the working copy of 'src/main/webapp/index.jsp', LF will be replaced
by CRLF the next time Git touches it

laksh@luxx MINGW64 ~/WebProject (main)
$ 

laksh@luxx MINGW64 ~/WebProject (main)
$ git add .
warning: in the working copy of 'src/main/webapp/WEB-INF/web.xml', LF will be re
placed by CRLF the next time Git touches it
warning: in the working copy of 'src/main/webapp/index.jsp', LF will be replaced
by CRLF the next time Git touches it

laksh@luxx MINGW64 ~/WebProject (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  .classpath
    new file:  .mvn/jvm.config
    new file:  .mvn/maven.config
    new file:  .project
    new file:  .settings/.jsdtscope
    new file:  .settings/org.eclipse.core.resources.prefs
    new file:  .settings/org.eclipse.jdt.core.prefs
    new file:  .settings/org.eclipse.m2e.core.prefs
    new file:  .settings/org.eclipse.wst.common.component
    new file:  .settings/org.eclipse.wst.common.project.facet.core.xml
    new file:  .settings/org.eclipse.wst.jsdt.ui.superType.container
    new file:  .settings/org.eclipse.wst.jsdt.ui.superType.name
    new file:  .settings/org.eclipse.wst.validation.prefs
    new file:  pom.xml
    new file:  src/main/webapp/WEB-INF/web.xml
    new file:  src/main/webapp/index.jsp
    new file:  target/m2e-wtp/web-resources/META-INF/MANIFEST.MF
    new file:  target/m2e-wtp/web-resources/META-INF/maven/com.app.demo/web
ProjectNew/pom.properties
    new file:  target/m2e-wtp/web-resources/META-INF/maven/com.app.demo/web
ProjectNew/pom.xml
```

Commit all the changes made to your Project:

```
laksh@luxx MINGW64 ~/WebProject (main)
$ git commit -m "WebCommit"
[main 3d65783] WebCommit
 19 files changed, 300 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .mvn/jvm.config
 create mode 100644 .mvn/maven.config
 create mode 100644 .project
 create mode 100644 .settings/.jsdtscope
 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.jsdt.ui.superType.container
 create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.name
 create mode 100644 .settings/org.eclipse.wst.validation.prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/webapp/WEB-INF/web.xml
 create mode 100644 src/main/webapp/index.jsp
 create mode 100644 target/m2e-wtp/web-resources/META-INF/MANIFEST.MF
 create mode 100644 target/m2e-wtp/web-resources/META-INF/maven/com.app.demo/WebProjectNew/pom.properties
 create mode 100644 target/m2e-wtp/web-resources/META-INF/maven/com.app.demo/WebProjectNew/pom.xml

laksh@luxx MINGW64 ~/WebProject (main)
$
```

Push the changes into GitHub:

```
laksh@luxx MINGW64 ~/WebProject (main)
$ git push
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Delta compression using up to 16 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (32/32), 4.50 KiB | 4.50 MiB/s, done.
Total 32 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/lxksh/WebProject.git
 e4ac6cf..3d65783  main -> main

laksh@luxx MINGW64 ~/WebProject (main)
$
```

Refresh GitHub to see The Updated Repository:

The screenshot shows a GitHub repository named 'WebProject'. The 'Code' tab is selected. The 'main' branch has 2 commits. The first commit is by 'Ivksh' with the message 'WebCommit' at 'now'. The second commit is an 'Initial commit' at '7 minutes ago'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and 'Create a new release'. The 'Packages' section says 'No packages published' and 'Publish your first package'.

3. INSTALLATION OF JENKINS

JENKINS INSTALLATION: DOWNLOAD JENKINS:

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

The Jenkins project produces two release lines: Stable (LTS) and weekly. Depending on your organization's needs, one may be preferred over the other. See the links below for more information and recommendations about the release lines.

Stable (LTS)

Long-Term Support (LTS) release baselines are chosen every 12 weeks from the stream of regular releases. Every 4 weeks we release stable releases which include bug and security fix backports. [Learn more...](#)

[Changelog](#) [Upgrade Guide](#) [Past Releases](#)

Weekly releases

This release line delivers bug fixes and new features rapidly to users and plugin developers who need them. It is generally delivered on a weekly cadence. [Learn more...](#)

[Changelog](#) [Past Releases](#)

Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the [Hardware and Software requirements](#) section of the User Handbook.
2. Select one of the packages below and follow the download instructions.

Click on windows application to download.

Download Jenkins 2.479.1 LTS for:

- Generic Java package (.war)**
SHA-256: dbf987b3aab16ce20e9413b3082fa323e3724cbb64562ddb64c1e4d4f58b470
- Docker**
- Kubernetes**
- Ubuntu/Debian**
- Red Hat/Fedora/Alma/Rocky/CentOS**
- Windows**

INSTALLATION OF JENKINS:

Jenkins 2.479.1 Setup

Welcome to the Jenkins 2.479.1 Setup Wizard

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



Back [Next](#) Cancel

Jenkins 2.479.1 Setup

Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install Jenkins 2.479.1 to:

C:\Program Files\Jenkins\

Change...



Back [Next](#) Cancel

Jenkins 2.479.1 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 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:

Port Selection

Choose a port for the service.

Please choose a port.

Port Number (1-65535):

Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.

Back Next Cancel Back Next Cancel

Jenkins 2.479.1 Setup

Select Java home directory (JDK or JRE)

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\

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.



The required Jenkins components

This feature requires 92MB on your hard drive. It has 2 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.

Back Next Cancel Reset Disk Usage Back Next Cancel

Launch local host 8080 on browser and follow the path shown:

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

.....

localhost:8080

Getting Started

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding
<input type="checkbox"/> Timestamper	<input type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input type="checkbox"/> Gradle
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline Graph View
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication
<input type="checkbox"/> LDAP	<input type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer	<input type="checkbox"/> Dark Theme

** Ionicons API
 Folders
 OWASP Markup Formatter
 ** ASM API

Jenkins 2.479.1

Jenkins

Dashboard > Manage Jenkins > Jenkins' own user database > Create User

Create User

Username

Password

Confirm password

Full name

E-mail address

Create User

Getting Started

Instance Configuration

Jenkins URL:

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 DASHBOARD:



Sign in to Jenkins

Username: Harshith

Password:

Keep me signed in

Sign in

localhost:8080

Jenkins

Search (CTRL+K)

Laksh Vijayvargiya log out

Dashboard >

+ New Item

Build History

Manage Jenkins

My Views

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

CLICK ON MANAGE JENKINS ON THE DASHBOARD:

localhost:8080/manage/

Jenkins

Search (CTRL+K)

Laksh Vijayvargiya log out

Dashboard > Manage Jenkins

+ New Item

Manage Jenkins

Search settings

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Set up agent **Set up cloud** **Dismiss**

System Configuration

Build Queue: No builds in the queue.

Build Executor Status: 0/2

Tools: Configure tools, their locations and automatic installers.

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

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

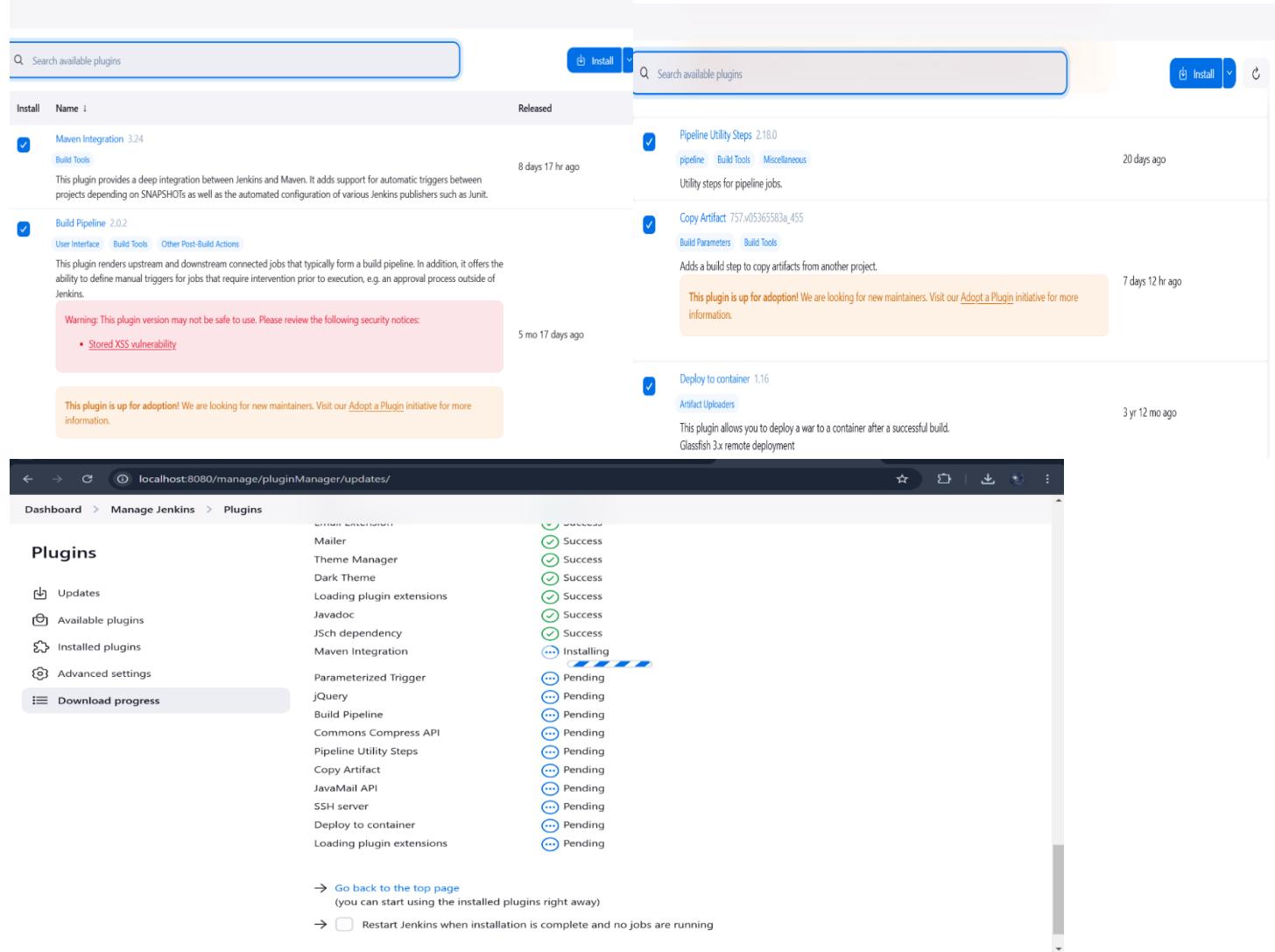
Clouds: Add, remove, and configure cloud instances to provision agents on-demand.

Appearance: Configure the look and feel of Jenkins

INSTALL PLUGINS IN JENKINS:

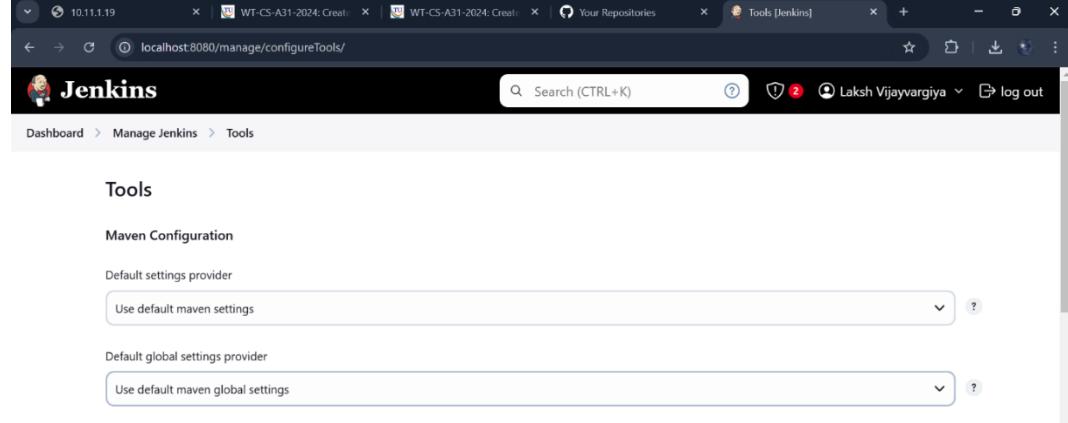
Check in the available and install the plugins
 Maven integration plugin,
 Build pipeline plugin,
 Pipeline utility,
 Copy artifacts,
 Deploy to container.

click on > install without restart.



The screenshot shows the Jenkins plugin manager interface. On the left, there's a sidebar with links: Dashboard, Manage Jenkins, Plugins, Updates, Available plugins, Installed plugins, Advanced settings, and Download progress. The main area has two sections: 'Available extensions' and 'Installed extensions'. Under 'Available extensions', several plugins are listed with their status: Mailer (Success), Theme Manager (Success), Dark Theme (Success), Loading plugin extensions (Success), Javadoc (Success), JSch dependency (Success), Maven Integration (Installing), Parameterized Trigger (Pending), jQuery (Pending), Build Pipeline (Pending), Commons Compress API (Pending), Pipeline Utility Steps (Pending), Copy Artifact (Pending), JavaMail API (Pending), SSH server (Pending), Deploy to container (Pending), and Loading plugin extensions (Pending). Below these lists are two buttons: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

Select Tools on Manage Jenkins Tab:



The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for 'Maven Configuration' and 'Default settings provider'. Under 'Maven Configuration', there are dropdown menus for 'Use default maven settings' and 'Use default maven global settings'. At the bottom of the page, there's a note: 'give the JAVA_HOME path:'.

give the JAVA_HOME path:

JDK installations

JDK installations ^ Edited

Add JDK

JDK

Name

JAVA_HOME

Install automatically

Add Installer

Add JDK

Git installations

Save

Apply

give the MAVEN_HOME path:

Maven installations ^ Edited

Add Maven

Maven

Name

MAVEN_HOME

Install automatically

Install from Apache

Version

3.9.9

Add Installer

Add Maven

Save

Apply

Experiment 6:

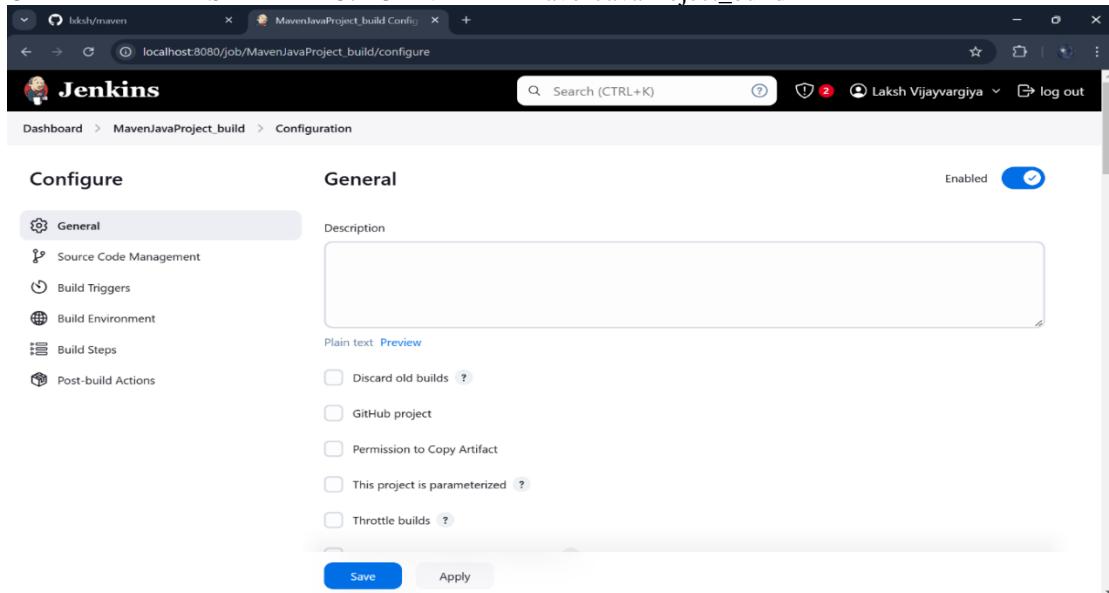
BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT

BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN WEB PROJECT WITH POLL SCM

BUILDING THE CI/CD SCRIPTED PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT WITH POLL SCM

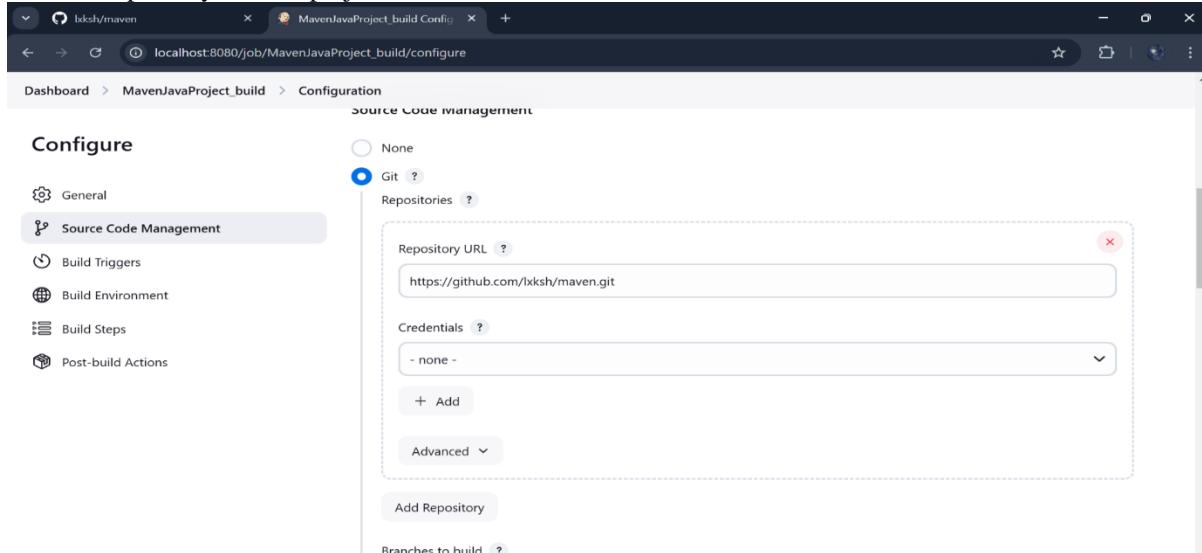
1. JENKINS FOR MAVEN JAVA PROJECT

CREATE A FREESTYLE PROJECT NAMED MavenJavaProject_build



The screenshot shows the Jenkins configuration interface for a job named "MavenJavaProject_build". The "General" tab is active. The "Description" field is empty. Under "Build Triggers", the "Discard old builds" option is checked. Under "Post-build Actions", there are several options: GitHub project, Permission to Copy Artifact, This project is parameterized, and Throttle builds. At the bottom are "Save" and "Apply" buttons.

Give Git repository URL of project:



The screenshot shows the Jenkins configuration interface for the same job. The "Source Code Management" tab is active. The "Git" option is selected, and the "Repository URL" field contains "https://github.com/lxksh/maven.git". The "Credentials" dropdown is set to "- none -". There are buttons for "+ Add" and "Advanced". Below the repository section, there is a "Branches to build" field.

Build and Invoke top-level Maven Targets:

Add maven_home with clean and install:

The screenshot shows the Jenkins configuration page for the 'MavenJavaProject_build' job. Under the 'Build Steps' section, there are two 'Invoke top-level Maven targets' configurations. The first step has a 'Goals' field containing 'clean'. The second step has a 'Goals' field containing 'install'. Both steps have 'MAVEN_HOME' selected in the 'Maven Version' dropdown. There are 'Save' and 'Apply' buttons at the bottom.

In post build actions-> select Archive the artifacts,to send output of build project to testing team , select archive “**” and in build other projects connect MavenJavaProject_test:

The screenshot shows the Jenkins configuration page for the 'MavenJavaProject_build' job. Under the 'Post-build Actions' section, there are two configurations: 'Archive the artifacts' and 'Build other projects'. The 'Archive the artifacts' step has a 'Files to archive' field containing '**'. The 'Build other projects' step has a 'Projects to build' field containing 'MavenJavaProject_test'. Underneath these, there are three trigger options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. There is also an 'Add post-build action' button. At the bottom are 'Save' and 'Apply' buttons.

Create a new freestyle project test :

The screenshot shows the Jenkins configuration interface for a job named 'MavenJavaProject_test'. The 'General' tab is selected. At the top right, there is an 'Enabled' switch with a checked checkbox. Below it, there is a large text area labeled 'Description' with a 'Plain text' link and a 'Preview' link. Underneath this are several checkboxes for build triggers and actions:

- Discard old builds
- GitHub project
- Permission to Copy Artifact
- This project is parameterized
- Throttle builds
- Execute concurrent builds if necessary

At the bottom of the screen are two buttons: 'Save' and 'Apply'.

We can mention git repository as None:

The screenshot shows the Jenkins configuration interface for a job named 'MavenJavaProject_test'. The 'Source Code Management' tab is selected. It contains a radio button group for 'None' (selected) and 'Git' (unchecked). Below this is a section for 'Build Triggers' with several checkboxes:

- Trigger builds remotely (e.g., from scripts)
- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling
- Poll SCM

Below the triggers is a section for 'Build Environment' with a checked checkbox for 'Delete workspace before build starts'. There is also an 'Advanced' dropdown and a checkbox for 'Use secret text(s) or file(s)'. At the bottom are 'Save' and 'Apply' buttons.

Add Build Steps and copy artifacts from build project:

The screenshot shows the Jenkins configuration interface for a job named 'MavenJavaProject_test'. The 'Build Steps' section is active, and a sub-step titled 'Copy artifacts from another project' is being configured. The configuration fields include:

- Project name:** MavenJavaProject_build
- Which build:** Latest successful build
- Stable build only:** Unchecked
- Artifacts to copy:** **
- Artifacts not to copy:** (empty)
- Target directory:** (empty)

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

Select Invoke top _ level Maven targets and add test in Goals:

The screenshot shows the Jenkins configuration interface for a job named 'MavenJavaProject_test'. The 'Build Steps' section is active, and a sub-step titled 'Invoke top-level Maven targets' is being configured. The configuration fields include:

- Maven Version:** MAVEN_HOME
- Goals:** test

Below the build steps, under 'Post-build Actions', there is a configuration for 'Archive the artifacts'.

In post build actions -> select Archive the artifacts:

The screenshot shows the Jenkins configuration interface for a job named 'MavenJavaProject_test'. Under 'Post-build Actions', a sub-action titled 'Archive the artifacts' is being configured. The configuration field includes:

- Files to archive:** **

Create a pipeline by clicking on + symbol in the dashboard ->a pipeline is a collection of events or jobs which are interlinked with one another in a sequence

The screenshot shows the Jenkins interface for the 'MavenJavaProject_build' job. On the left, there's a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. Below that is a 'Builds' section showing build history from #8 down to #3. The main content area has a title 'MavenJavaProject_build'. It displays 'Last Successful Artifacts' (including jvm.config, maven.config, org.eclipse.core.resources_prefs, org.eclipse.jdt.core_prefs, org.eclipse.m2e.core_prefs, pom.xml, App.java, AppTest.java, App.class, pom.properties, default-compile/createdFiles.lst, default-compile/inputFiles.lst, default-testCompile/createdFiles.lst, default-testCompile/inputFiles.lst, mavenjavalexample-0.0.1-SNAPSHOT.jar, com.demo.app.mavenjavalexample.AppTest.txt, TEST-com.demo.app.mavenjavalexample.AppTest.xml, and AppTest.class) and a 'Downstream Projects' section with 'MavenJavaProject_test'. There's also a 'Permalinks' section.

name to the pipeline->select Build Pipeline View->create

The screenshot shows the Jenkins 'New view' creation page. On the left, there's a sidebar with 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. Below that is a 'Build Queue' section indicating 'No builds in the queue.' and a 'Build Executor Status' section showing '0/2'. The main area is titled 'New view' and has a 'Name' field containing 'MavenJavaProject_pipe'. Under 'Type', 'Build Pipeline View' is selected (indicated by a blue circle). A description below it says: '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.' There are also 'List View' and 'My View' options. At the bottom is a 'Create' button. At the very bottom right, it says 'REST API' and 'Jenkins 2.479.1'.

Select the first project to trigger the execution->build project:

The image shows two screenshots of the Jenkins interface. The top screenshot is a configuration page for a pipeline view named 'mavenjava'. It includes sections for 'Build Pipeline View Title' (empty), 'Pipeline Flow' (Layout: 'Based on upstream/downstream relationship'), and 'Trigger Options' (Build Cards dialog open with 'OK' and 'Apply' buttons). The bottom screenshot is the Jenkins dashboard showing the 'MavenJavaProject_pipeline' view. It displays two jobs: 'MavenJavaProject_build' (last success: 2 min 33 sec ago, last failure: 35 min ago, duration: 8.8 sec) and 'MavenJavaProject_test' (last success: 2 min 18 sec ago, last failure: 13 min ago, duration: 4.2 sec). The dashboard also shows an empty 'Build Queue' and 'Build Executor Status' (0/2).

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 ?

MavenJavaProject_build

Trigger Options

Build Cards

OK Apply

Jenkins

Dashboard

All MavenJavaProject_pipeline +

S	W	Name	Last Success	Last Failure	Last Duration
Green checkmark	Sunny icon	MavenJavaProject_build	2 min 33 sec #8	35 min #4	8.8 sec
Green checkmark	Sunny icon	MavenJavaProject_test	2 min 18 sec #4	13 min #1	4.2 sec

Build Queue

No builds in the queue.

Build Executor Status

Icon: S M L

REST API Jenkins 2.479.1

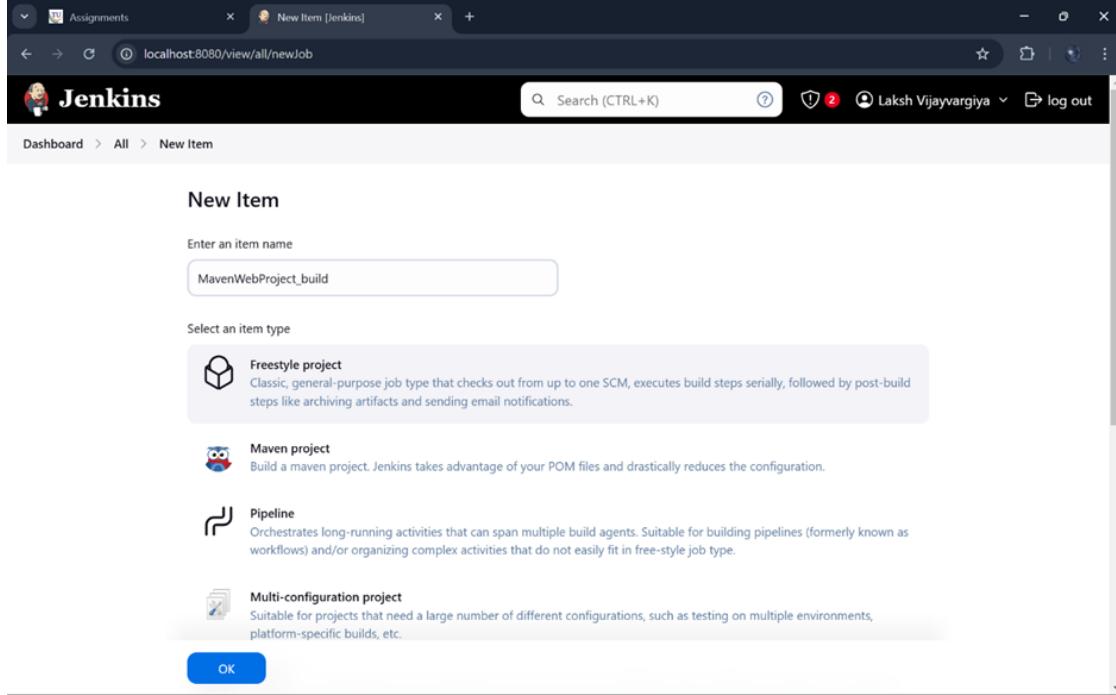
The pipeline is successful if it is in green color as shown ->check the console of the test project:

The screenshot shows the Jenkins Build Pipeline interface. At the top, there are tabs for 'lksh/maven' and 'MavenJavaProject_pipeline [Test]'. The main title is 'Build Pipeline'. Below the title are buttons for 'Run', 'History', 'Configure', 'Add Step', 'Delete', and 'Manage'. On the left, a sidebar shows a 'Pipeline' section with '#8'. In the center, two green boxes represent successful builds: '#8 MavenJavaProject_build' (16-Nov-2024 11:09:04 am, 8.8 sec, by 'laksh') and '#4 MavenJavaProject_test' (16-Nov-2024 11:09:19 am, 4.2 sec). An orange bar at the bottom spans the width of the green boxes.

The screenshot shows the Jenkins job details for 'MavenJavaProject_test'. The top navigation bar includes 'lksh/maven', 'MavenJavaProject_pipeline [Test]', and 'localhost:8080/job/MavenJavaProject_test/'. The main title is 'MavenJavaProject_test'. On the left, a sidebar has options: 'Status' (highlighted), 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. Below this is a 'Builds' section with a 'Filter' input and a list of builds: #4 (11:09 am), #3 (11:04 am), #2 (11:01 am), and #1 (10:58 am). The main content area shows 'Last Successful Artifacts' with a list of files and their sizes: jvm.config (0 B), maven.config (0 B), org.eclipse.core.resources.prefs (119 B), org.eclipse.jdt.core.prefs (425 B), org.eclipse.m2e.core.prefs (90 B), pom.xml (3.13 KiB), App.java (190 B), AppTest.java (342 B), App.class (573 B), pom.properties (75 B), default-compile/createdFiles.lst (40 B), default-compile/inputFiles.lst (118 B), default-testCompile/createdFiles.lst (44 B), default-testCompile/inputFiles.lst (122 B), mavenjavaexample-0.0.1-SNAPSHOT.jar (2.87 KiB), com.demo.app.mavenjavaexample.AppTest.txt (328 B), TEST-com.demo.app.mavenjavaexample.AppTest.xml (6.47 KiB), and AppTest.class (533 B). At the bottom, there is an 'Upstream Projects' section and a footer with the URL 'localhost:8080/job/MavenJavaProject_test/lastSuccessfulBuild/.../pom.prop'.

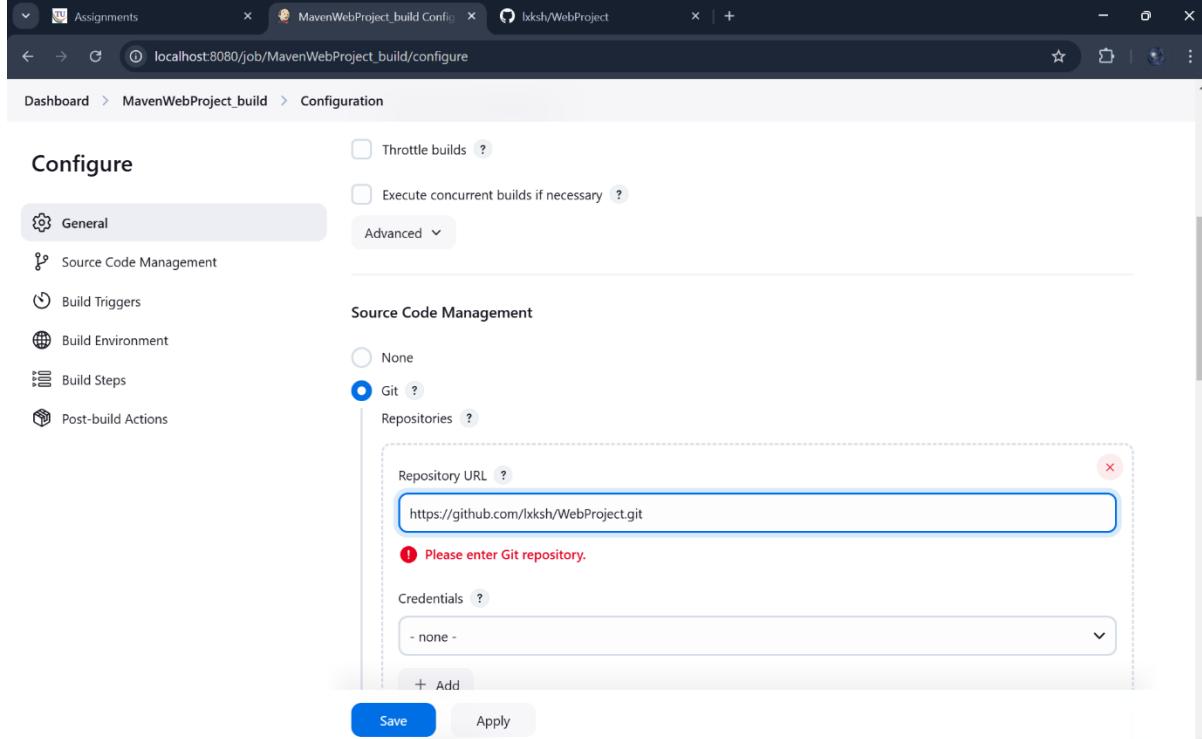
2. JENKINS WEB PROJECT

CREATE A FREESTYLE PROJECT NAMED MavenWebProject_build



The screenshot shows the Jenkins 'New Item' creation interface. In the 'Enter an item name' field, 'MavenWebProject_build' is typed. Under 'Select an item type', the 'Freestyle project' option is selected, described as a 'Classic, general-purpose job type'. Other options like 'Maven project', 'Pipeline', and 'Multi-configuration project' are also listed. At the bottom right is an 'OK' button.

Give Git repository URL of project:



The screenshot shows the Jenkins 'Configure' screen for the 'MavenWebProject_build' job. Under the 'General' tab, the 'Source Code Management' section is active, showing 'Git' selected as the provider. The 'Repository URL' field contains 'https://github.com/lksh/WebProject.git', which is highlighted with a blue border. A red error message 'Please enter Git repository.' is displayed below the field. The 'Credentials' dropdown is set to '- none -'. At the bottom are 'Save' and 'Apply' buttons.

Build triggers

Build Environment

Build Steps

Post-build Actions

Branches to build

Branch Specifier (blank for 'any') ?

Add Branch

Repository browser ?

Additional Behaviours

Add

Add * * * * in Poll SCM:

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, 22 November, 2024, 3:43:39 pm India Standard Time; would next run at Friday, 22 November, 2024, 3:43:39 pm India Standard Time.

Build and Invoke top-level Maven Targets:

Add maven_home with clean and install:

Assignments MavenWebProject_build Config lksh/WebProject +

localhost:8080/job/MavenWebProject_build/configure

Dashboard > MavenWebProject_build > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Maven Version

Goals

Advanced

Invoke top-level Maven targets

Maven Version

Goals

Advanced

In post build actions-> select Archive the artifacts,to send output of build project to testing team , select archive “**/*” and in build other projects connect MavenWebProject_test :

The screenshot shows the Jenkins configuration interface for a job. The left sidebar has tabs: Build Triggers, Build Environment, Build Steps, and Post-build Actions. The Post-build Actions tab is selected. Under Post-build Actions, there are two sections: "Archive the artifacts" and "Build other projects".

- Archive the artifacts:** A dropdown menu "Files to archive" contains the pattern "**/*". There is an "Advanced" button.
- Build other projects:** A dropdown menu "Projects to build" contains "MavenWebProject_test". A message says "No such project 'MavenWebProject_test'. Did you mean 'MavenJavaProject_test'?".
 - Trigger only if build is stable
 - Trigger even if the build is unstable
 - Trigger even if the build fails

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

The screenshot shows the Jenkins dashboard for the "MavenWebProject_build" job. The top navigation bar includes links for Assignments, MavenWebProject_build [Jenkins], and lkshv/WebProject. The main content area shows the job name "MavenWebProject_build" and a "Status" card. The left sidebar has options: Changes, Workspace, Build Now, Configure, Delete Project, Git Polling Log (which is selected), and Rename.

Permalinks:

- Last build (#1), 16 sec ago

Builds: A table with one row: #1 3:45pm. It includes a "Filter" input field and a "Today" link.

At the bottom, the URL is localhost:8080/job/MavenWebProject_build/scmPoll.log, and the page footer shows REST API and Jenkins 2.479.1.

Create a new freestyle project test :

The screenshot shows the Jenkins configuration interface for a job named 'MavenJavaProject_test'. The 'General' tab is selected. At the top right, there is an 'Enabled' switch which is turned on. Below it, there is a large text area labeled 'Description' with a placeholder 'Plain text' and a 'Preview' link. Underneath this are several checkboxes for build triggers and options:

- Discard old builds
- GitHub project
- Permission to Copy Artifact
- This project is parameterized
- Throttle builds
- Execute concurrent builds if necessary

At the bottom of the screen are two buttons: 'Save' and 'Apply'.

Delete workspace before build starts:

The 'Build Environment' tab is selected. Under the 'Build Steps' section, there is a checkbox labeled 'Delete workspace before build starts' which is checked. Below it is an 'Advanced' dropdown menu with several other options:

- 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

At the bottom of the screen are two buttons: 'Save' and 'Apply'.

Add Build Steps and copy artifacts from build project:

The screenshot shows the Jenkins configuration interface for a job named 'MavenWebProject_test'. The 'Build Steps' section is selected. A sub-section titled 'Copy artifacts from another project' is open. It includes fields for 'Project name' (set to 'MavenWebProject_build'), 'Which build' (set to 'latest successful build'), and 'Artifacts to copy' (set to '**/*'). There are also fields for 'Artifacts not to copy' and 'Target directory'. At the bottom are 'Save' and 'Apply' buttons.

Select Invoke top _ level Maven targets and add test in Goals :

The screenshot shows the Jenkins configuration interface for a job named 'MavenWebProject_test'. The 'Post-build Actions' section is selected. A sub-section titled 'Invoke top-level Maven targets' is open. It includes fields for 'Maven Version' (set to 'MAVEN_HOME') and 'Goals' (set to 'test'). There are also checkboxes for 'Flatten directories', 'Optional', 'Fingerprint Artifacts', and 'Include Build Number'. At the bottom are 'Save' and 'Apply' buttons.

In post build actions -> select Archive the artifacts :

The screenshot shows the Jenkins configuration interface for a job named 'MavenWebProject_test'. The 'Post-build Actions' section is selected. A sub-section titled 'Archive the artifacts' is open. It includes a field for 'Files to archive' (set to '**/*'). There are also checkboxes for 'Advanced' and 'Add post-build action'. At the bottom are 'Save' and 'Apply' buttons.

Build Environment

Build Steps

Post-build Actions

Build other projects

Projects to build: MavenWebProject_deploy

No such project 'MavenWebProject_deploy'. Did you mean 'MavenWebProject_build'?

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**

Create new freestyle project:

Assignments × New Item [Jenkins] × localhost:8080/view/all/newJob × Ixksh/WebProject × +

Laksh Vijayvargiya log out

Jenkins

Dashboard > All > New Item

New Item

Enter an item name: MavenWebProject_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.

OK

Add Build Steps and copy artifacts from build project :P

Assignments × MavenWebProject_deploy Config × localhost:8080/job/MavenWebProject_deploy/configure × Ixksh/WebProject × +

Dashboard > MavenWebProject_deploy > Configuration

Configure

Build Steps

Copy artifacts from another project

Project name: MavenWebProject_test

Which build: Latest successful build

Stable build only

Artifacts to copy: *

Artifacts not to copy:

Target directory:

Save **Apply**

Create war files and add tomcat in containers:

The screenshot shows the Jenkins job configuration page for 'MavenWebProject_deploy'. The left sidebar has 'Post-build Actions' selected. The main area is titled 'Deploy war/ear to a container'. It includes fields for 'WAR/EAR files' (set to '**.war'), 'Context path' (set to 'projecttest'), and a 'Containers' section for 'Tomcat 9.x Remote'. The 'Credentials' dropdown is set to '- none -', and the 'Tomcat URL' field contains 'http://localhost:8083/'. At the bottom are 'Save' and 'Apply' buttons.

Add jenkins credentials:

The screenshot shows the Jenkins 'Configure Jenkins Credentials Provider: Jenkins' page. Under 'Add Credentials', it shows a 'Domain' of 'Global credentials (unrestricted)', 'Kind' of 'Username with password', and 'Scope' of 'Global (Jenkins, nodes, items, all child items, etc.)'. It includes fields for 'Username' and 'Password', with an unchecked 'Treat username as secret' checkbox. Below this, the Jenkins dashboard shows the 'Configure' screen for the 'MavenWebProject_deploy' job, specifically the 'Post-build Actions' section. It shows the 'Tomcat 9.x Remote' configuration with 'Tomcat URL' set to 'http://localhost:8083/' and other options like 'Advanced' and 'Add Container'.

Create a pipeline by clicking on + symbol in the dashboard ->a pipeline is a collection of events or jobs which are interlinked with one another in a sequence

localhost:8080/newView

New view

Name MavenWebProject_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

Select the first project to trigger the execution->build project:

localhost:8080/view/MavenWebProject_pipeline/configure

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 ? MavenWebProject_build

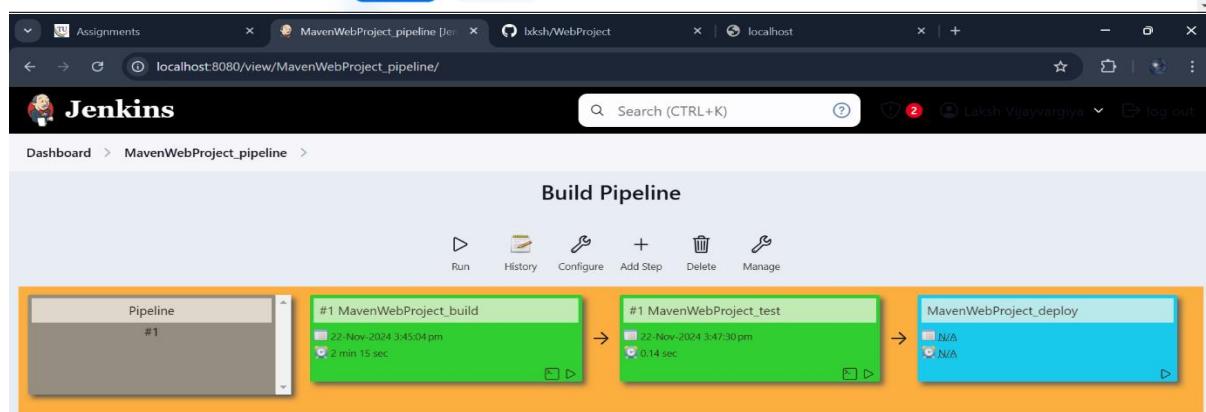
Trigger Options

Build Cards Standard build card

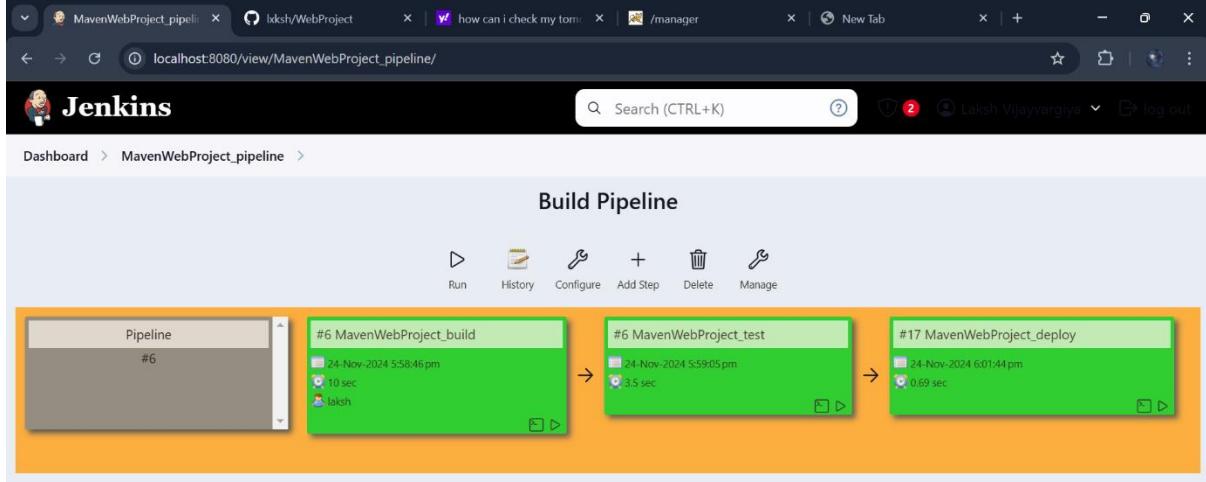
Use the default build cards

Restrict triggers to most recent successful builds ?

OK **Apply**



The pipeline is successful if it is in green color as shown ->check the console of the test project



Here go to projecttest and you can see the output:

The Tomcat Manager Applications page lists several deployed 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
/WebProjectNew	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/projecttest	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy
Deploy directory or WAR file located on server

Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy

WAR file to deploy

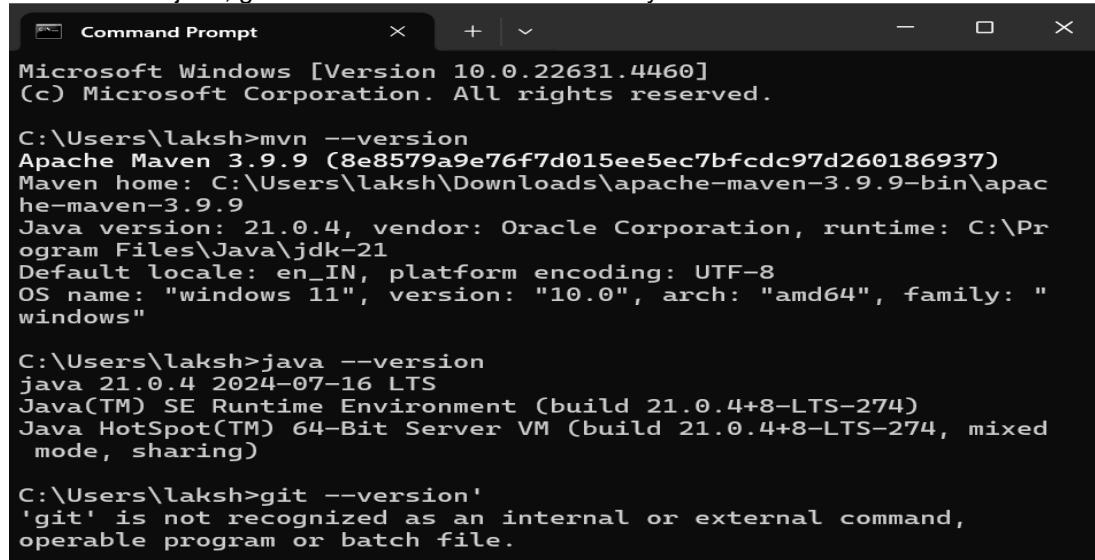
localhost:8083/projecttest/

Hello World! from Laksh

3.SCRIPTED

Sample hello script

Check for the java, git and maven version whether they are installed



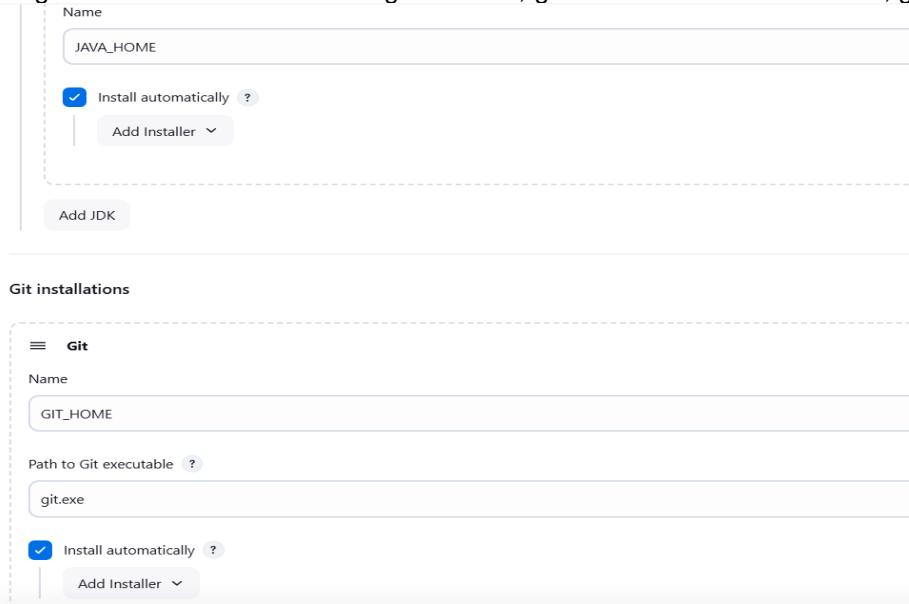
```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\laksh>mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfc97d260186937)
Maven home: C:\Users\laksh\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9
Java version: 21.0.4, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"

C:\Users\laksh>java --version
java 21.0.4 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 21.0.4+8-LTS-274)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.4+8-LTS-274, mixed mode, sharing)

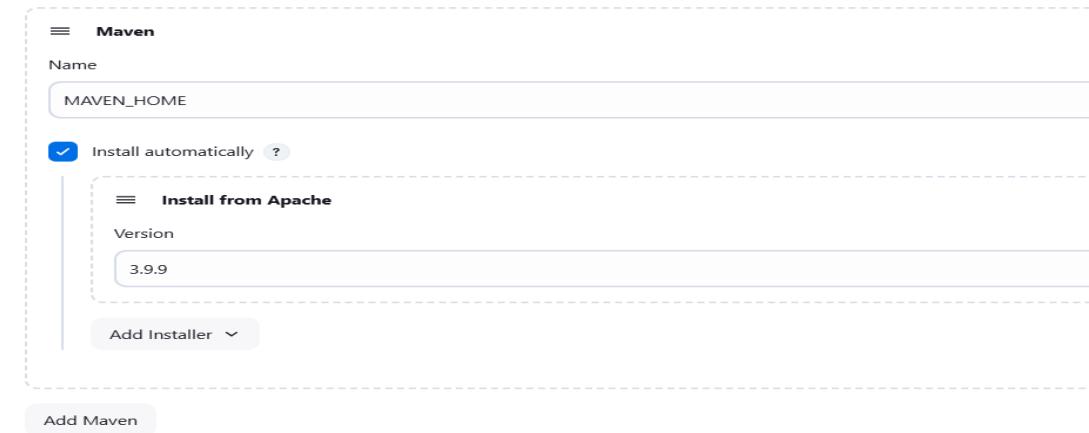
C:\Users\laksh>git --version
'git' is not recognized as an internal or external command,
operable program or batch file.
```

Go to manage Jenkins and click on manage Jenkins , go to tools and enable maven, git , jdk



The screenshot shows the Jenkins configuration interface for managing tools. It includes sections for Java, Git, and Maven installations, each with fields for Name, Path to executable, and an 'Install automatically' checkbox.

Tool	Name	Path to Executable	Install Automatically
Java	JAVA_HOME		<input checked="" type="checkbox"/> Install automatically
Git	GIT_HOME	git.exe	<input checked="" type="checkbox"/> Install automatically
Maven			



Come to dashboard and click on the new item and name it as script and select pipeline and click ok
New Item

Enter an item name

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**
Available for projects that need a large number of different configurations, such as testing on multiple environments.

OK

Now in configuration go to pipeline and enable pipeline script , drag on the hello world and change the sample hello world script program

Pipeline script

Script ?

```

1 ▾ pipeline {
2     agent any
3
4 ▾   stages {
5     ▾       stage('Hello1') {
6         ▾           steps {
7             echo 'Hello World1'
8         }
9     }
10    ▾       stage('Hello2') {
11        ▾           steps {
12            echo 'Hello World2'
13        }
14    }
15  }
16 }
17

```

Apply and save it

Dashboard > sampleScript >

Build Now Stage view

Configure Delete Pipeline Full Stage View Stages Rename Pipeline Syntax

Builds Filter

Today
Nov 21 15:33 #2
20 November 2024
Nov 20 12:51 #1

Average stage times:
(Average full run time: ~3s)

Stage view

	Hello1	Hello2
#2 Nov 21 15:33 No Changes	161ms	98ms
#1 Nov 20 12:51 No Changes	117ms	91ms
	206ms	106ms

Permalinks

- Last build (#1), 1 day 2 hr ago
- Last stable build (#1), 1 day 2 hr ago
- Last successful build (#1), 1 day 2 hr ago
- Last completed build (#1), 1 day 2 hr ago

Now build it , u will observe the stage view of it on the right

		sampleMavenWeb_deploy	23 hr	#13	23 h #
		sampleMavenWeb_test	23 hr	#5	N/A
		sampleScript	55 sec	#2	N/A

Icon: S M L

Declarative script

Create a declarative script in new item and change the pipeline script as needed for you

Change the script to function clean, install, test and download package

Pipeline

Definition

Pipeline script

Script ?

```
1 ~ pipeline {
2   agent any
3   tools {
4     maven 'MAVEN_HOME'
5     git 'GIT_HOME'
6   }
7   stages {
8     stage('Git Repository & clean') {
9       steps {
10       bat 'rmdir /q Maven_demo_3'
11       bat 'git clone https://github.com/rishith-nallapu/Maven_demo_3.git'
12       bat 'mvn clean -f Maven_demo_3'
13     }
14   }
15   stage('Install') {
16     steps {
17       bat 'mvn install -f Maven_demo_3'
```

Pipeline

Definition

Pipeline script

Script ?

```
14 ~
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30 }
```

```
stage('Install') {
  steps {
    bat 'mvn install -f Maven_demo_3'
}
stage('Test') {
  steps {
    bat 'mvn test -f Maven_demo_3'
}
stage('Package') {
  steps {
    bat 'mvn package -f Maven_demo_3'
```

Use Groovy Sandbox ?

Save

Apply

Click on build now and observe the staging view on the right

Dashboard > declarativescript >

Status

declarativescript

Add description

</> Changes

▷ Build Now

⚙ Configure

>Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Builds

Stage View

Declarative: Tool Install	Git Repository & Clean	Install			Test		Package	
		188ms	5s	8s	5s	5s	5s	5s
#13 Nov 20 14:42 No Changes	201ms	4s	6s	5s	5s	5s	5s	5s
#12 Nov 20 14:41 No Changes	232ms	6s	6s	4s	5s	5s	5s	5s
#11 Nov 20 14:40 No Changes	142ms	4s	6s	4s	5s	5s	5s	5s

Our stages are build

Experiment 7:

INSTALLATION OF DOCKER, MINIKUBE, ACCOUNT IN DOCKERHUB

DOCKER CLI COMMANDS

MODIFY AND PUSH DOCKER IMAGE

CREATE AND PUSH DOCKERFILE IMAGE

RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE

DEPLOYING AND SCALING APPLICATIONS USING MINIKUBE

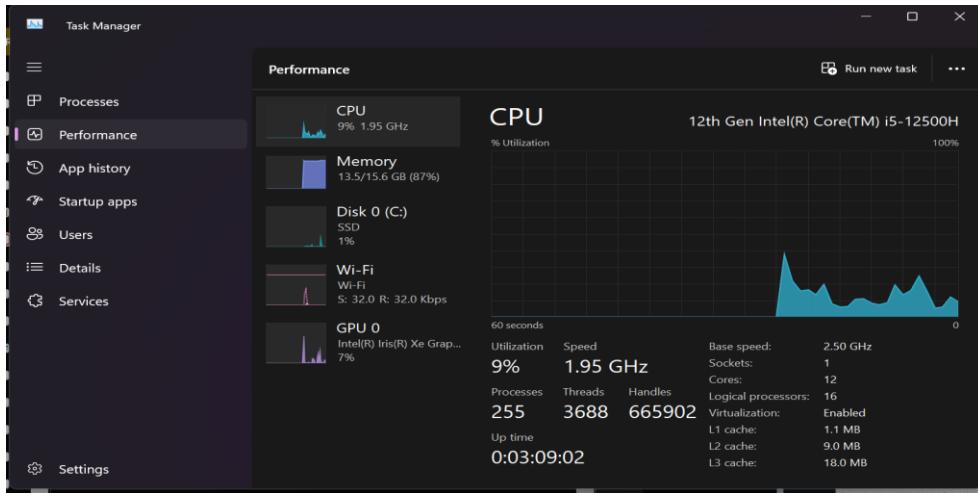
DEPLOYING AND MANAGING MONITORING SYSTEMS USING NAGIOS IN DOCKER

1. INSTALLATIONS

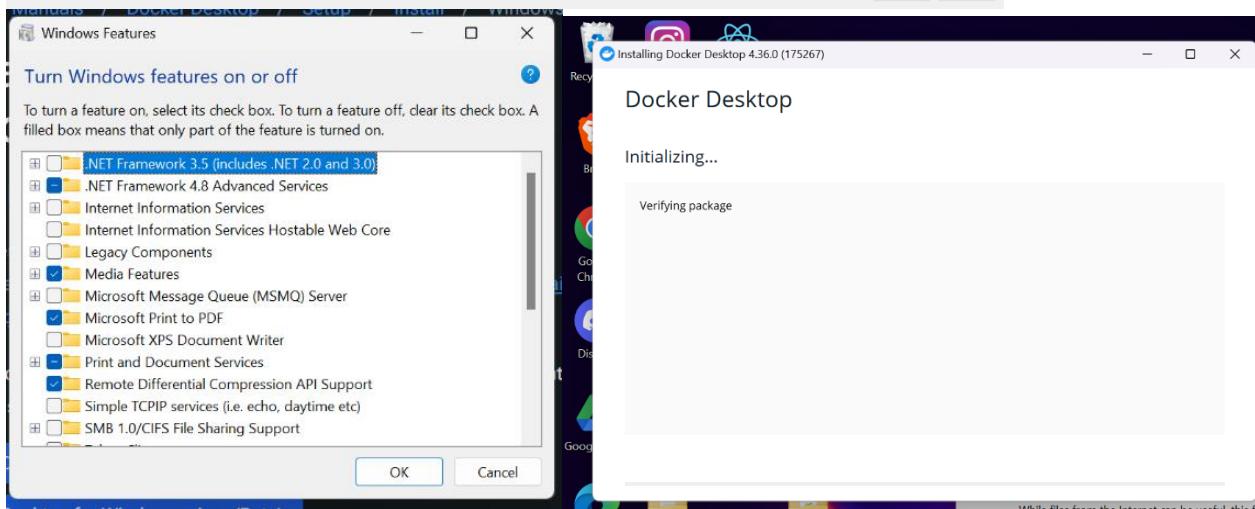
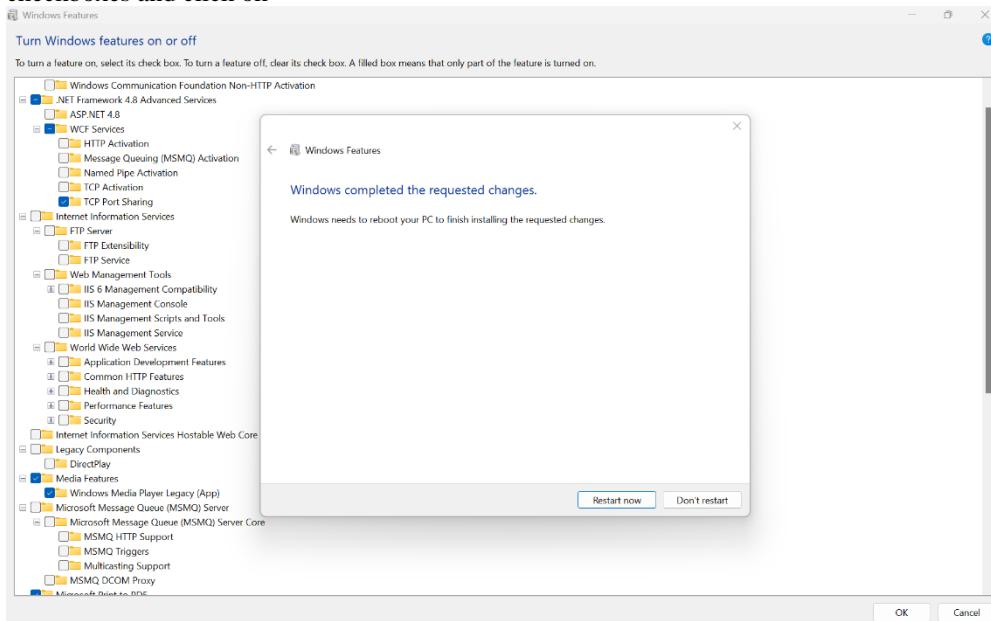
- 1. Docker Desktop for Windows
- 2. WSL installation if needed
- 3. Account in Docker Hub
- 4. Minikube (lightweight version of Kubernetes) installation

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Install Docker Desktop on Windows'. The page content includes a sidebar with navigation links for Docker Build, Docker Build Cloud, Docker Compose, Docker Desktop, Setup, Install, Mac, and Windows. The 'Windows' link is highlighted. The main content area features a title 'Install Docker Desktop on Windows', a section about Docker Desktop terms, and download links for 'Docker Desktop for Windows - x86_64' and 'Docker Desktop for Windows - Arm (Beta)'. A note at the bottom says 'For checksums, see [Release notes](#)'. On the right side, there's a 'Table of contents' section with links to system requirements, installation methods, and next steps. At the bottom, a cookie consent banner asks for permission to store cookies.

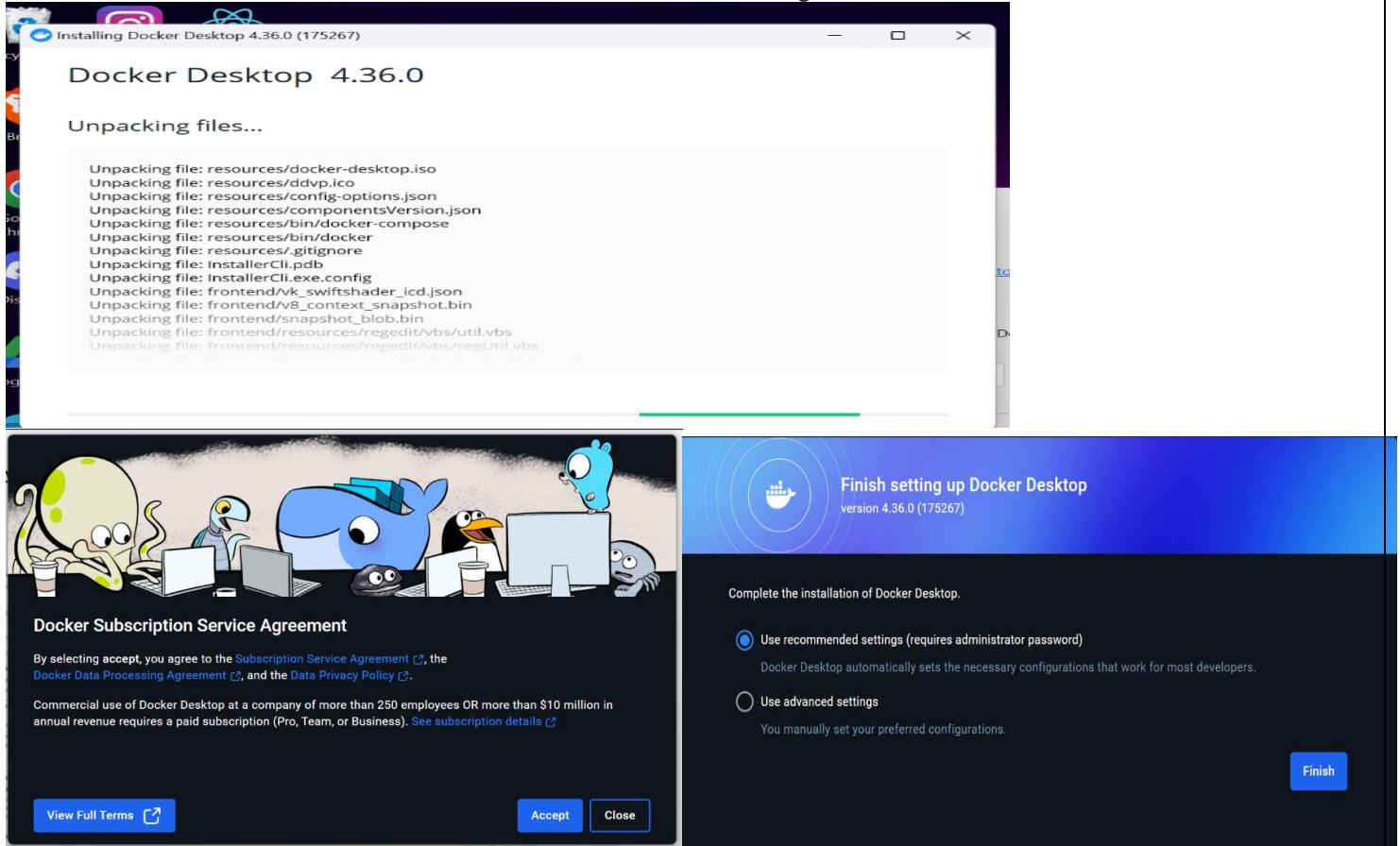
Make sure Virtualization is enabled in the system, we can check it in task manager as shown



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



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



Installation of wsl:

```
luxx@luxx: ~      x + v
Retype new password:
No password has been supplied.
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
No password has been supplied.
New password:
Retype new password:
No password has been supplied.
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

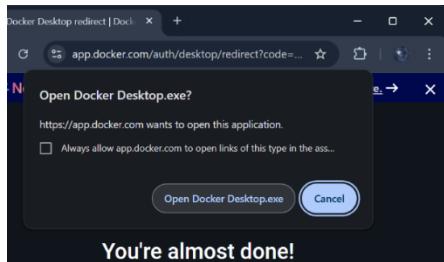
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

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

This message is shown once a day. To disable it please create the
/home/luxx/.hushlogin file.
luxx@luxx:~$ |
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

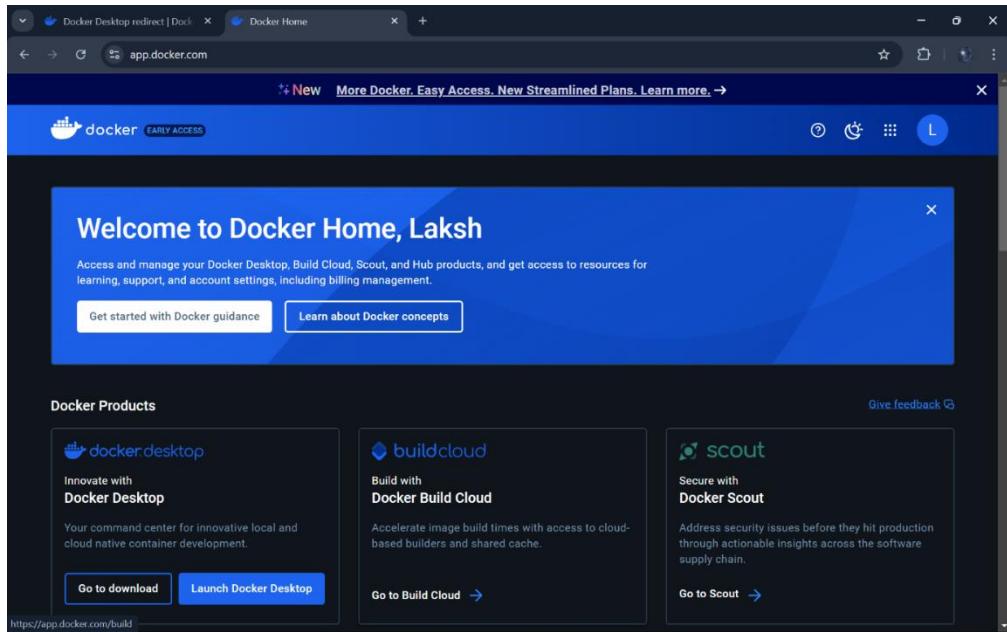
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
The operation completed successfully.
PS C:\Windows\system32>
```



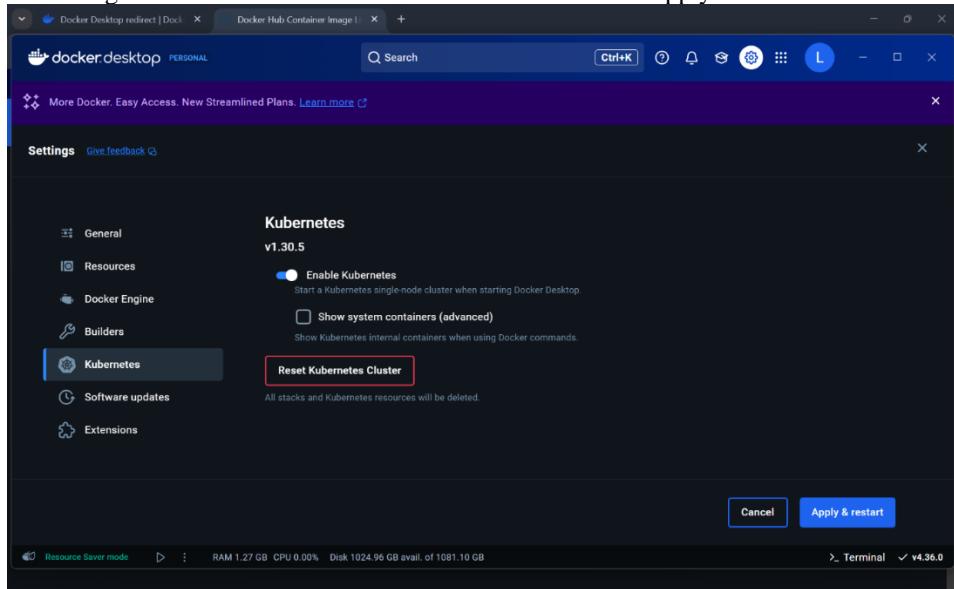
Create Docker id and login into docker desktop

Sign up in hub.docker.com

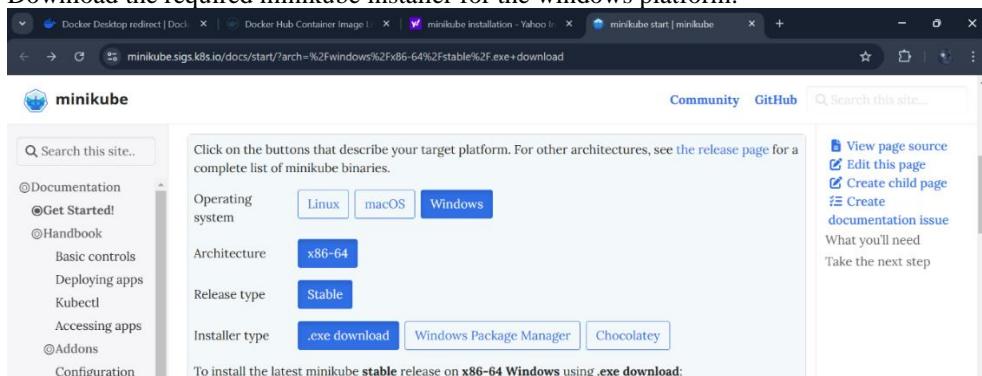


Install using Docker Desktop

Settings → Kubernetes → Check Enable Kubernetes → Apply

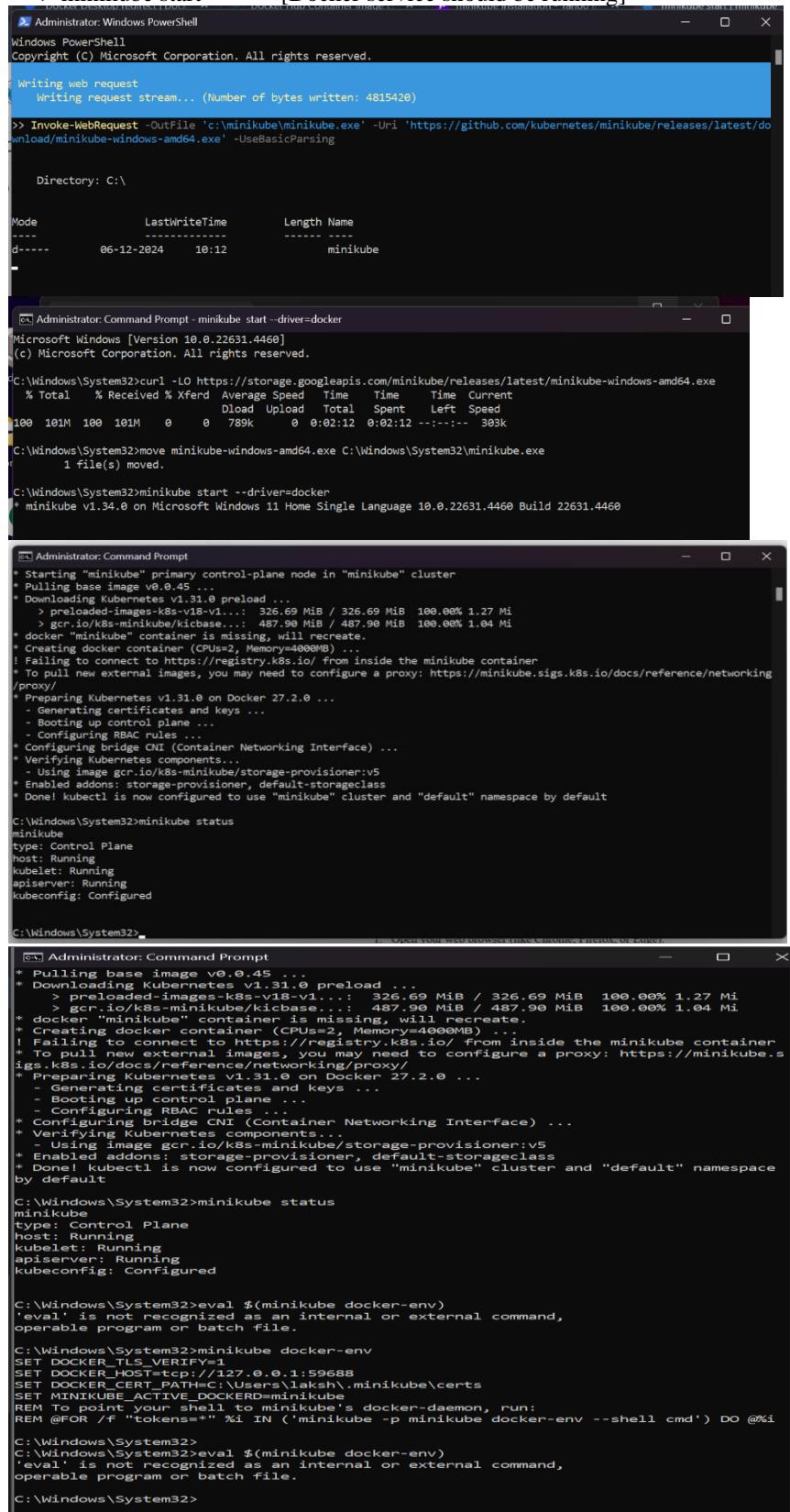


[Minikube](#) is a tool that allows you to run a single-node Kubernetes cluster inside a virtual machine, on your own computer. Download the required minikube installer for the windows platform.



Start your cluster by running the following command in powershell

> minikube start [Docker service should be running]



The image contains four screenshots of Windows Command Prompt windows, each showing a different step in the process of starting a minikube cluster:

- Screenshot 1:** Administrator: Windows PowerShell window. It shows the command "minikube start" being run, followed by output indicating a web request is being written to file "c:\minikube\minikube.exe". The directory listing shows a single file named "minikube".
- Screenshot 2:** Administrator: Command Prompt window. It shows the curl command to download "minikube-windows-amd64.exe" from the GitHub releases page. It also shows the move command to rename the downloaded file to "minikube.exe".
- Screenshot 3:** Administrator: Command Prompt window. It shows the "minikube start --driver=docker" command being run, which outputs the version information "minikube v1.34.0" and the message "Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default".
- Screenshot 4:** Administrator: Command Prompt window. It shows the "minikube status" command being run, which outputs the cluster status: "type: Control Plane", "host: Running", "kublet: Running", "apiserver: Running", and "kubeconfig: Configured".

2.DOCKER COMMANDS

Check docker version using docker –version

Logging into docker

Using docker pull to pull and hello-world image from docker hub

The left screenshot shows the output of 'docker --version' command:

```
PS C:\Windows\system32> docker --version
Docker version 27.3.1, build ce12230
```

The right screenshot shows the output of 'docker pull hello-world' command:

```
PS C:\Windows\system32> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fc966
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
PS C:\Windows\system32>
```

Running the hello-world image using docker run command

```
PS C:\Windows\system32> docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

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

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

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

For more examples and ideas, visit:
https://docs.docker.com/get-started/
PS C:\Windows\system32>
```

Lists all running Docker containers using docker ps

Lists all Docker containers, including running, stopped, and exited ones using docker ps -a.

Removing a stopped Docker container using docker rm.

Removing a Docker image from the local system using docker rmi.

Downloading a Docker image from docker hub using docker pull.

```
PS C:\Windows\system32> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
5e4142c429b1        hello-world        "/hello"           26 seconds ago    Exited (0) 24 seconds ago
                  distracted_heisenberg
bc81bd6703d8        gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr..."   14 minutes ago    Up 14 minutes
   127.0.0.1:59687->22/tcp, 127.0.0.1:59688->2376/tcp, 127.0.0.1:59690->5000/tcp, 127.0.0.1:59691->8443/tcp, 127.0.0.1:59692->32443/tcp
   minikube
PS C:\Windows\system32> docker rm 5e4142c429b1
5e4142c429b1
PS C:\Windows\system32> docker rmi hello-world
Untagged: hello-world:latest
Deleted: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fc966
PS C:\Windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8242f9d5b464: Download complete
75dfffa679c9b: Download complete
bc0965b23a04: Download complete
4f4fb700ef54: Download complete
9501a6ec095f: Download complete
98e7597530ef: Download complete
141f00dd5fee8: Download complete
8912a88e73c8: Download complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Windows\system32>
```

Running a new container named newredis in detached mode using the redis image.

Starting an interactive session inside the newredis container, running the redis command like set and get.

```
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Windows\system32> docker run --name my-redis -d redis
a0dd1a7cba346ee267ded6269338d282636ffd4ff3fc95ab5c903b021b9946a4
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE          COMMAND           CREATED          STATUS          PORTS
 NAMES
a0dd1a7cba34   redis          "docker-entrypoint.s..."   7 seconds ago   Up 7 seconds   6379/tcp

      my-redis
bc81bd6703d8   gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr..."   19 minutes ago   Up 19 minutes   127.0.0.1:59687->222/tcp, 127.0.0.1:59688->2376/tcp, 127.0.0.1:59690->5000/tcp, 127.0.0.1:59691->8443/tcp, 127.0.0.1:59689->32443/tcp   minikube
PS C:\Windows\system32> docker exec -it my-redis redis-cli
127.0.0.1:6379> SET name "Laksh"
OK
127.0.0.1:6379> GET name
"Laksh"
127.0.0.1:6379>
```

Stopping the newredis container that is running.

```
PS C:\Windows\system32> docker exec -it my-redis redis-cli
127.0.0.1:6379> SET name "Laksh"
OK
127.0.0.1:6379> GET name
"Laksh"
127.0.0.1:6379> docker stop my-redis
(error) ERR unknown command 'docker', with args beginning with: 'stop' 'my-redis'
127.0.0.1:6379> stop my-redis
(error) ERR unknown command 'stop', with args beginning with: 'my-redis'
127.0.0.1:6379>
PS C:\Windows\system32> docker stop my-redis
my-redis
PS C:\Windows\system32> docker start my-redis
my-redis
PS C:\Windows\system32>
```

Starting the newredis container.

Deleting an container using the command docker rm containerID and docker image using docker rmi.

```
Administrator: Windows PowerShell

      NAMES
a0dd1a7cba34   redis          "docker-entrypoint.s..."   7 seconds ago   Up 7 seconds   6379/tcp

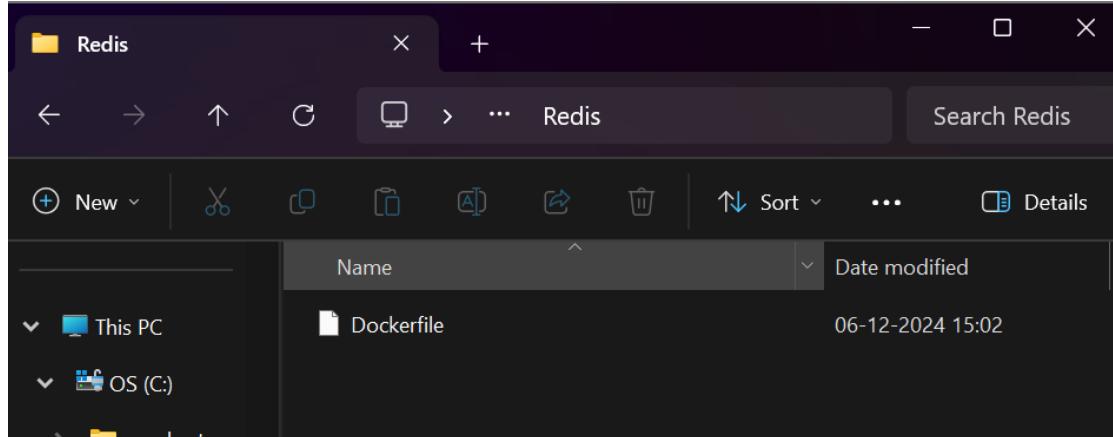
      my-redis
bc81bd6703d8   gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr..."   19 minutes ago   Up 19 minutes   127.0.0.1:59687->222/tcp, 127.0.0.1:59688->2376/tcp, 127.0.0.1:59690->5000/tcp, 127.0.0.1:59691->8443/tcp, 127.0.0.1:59689->32443/tcp   minikube
PS C:\Windows\system32> docker exec -it my-redis redis-cli
127.0.0.1:6379> SET name "Laksh"
OK
127.0.0.1:6379> GET name
"Laksh"
127.0.0.1:6379> docker stop my-redis
(error) ERR unknown command 'docker', with args beginning with: 'stop' 'my-redis'
127.0.0.1:6379> stop my-redis
(error) ERR unknown command 'stop', with args beginning with: 'my-redis'
127.0.0.1:6379>
PS C:\Windows\system32> docker stop my-redis
my-redis
PS C:\Windows\system32> docker start my-redis
my-redis
PS C:\Windows\system32> docker stop my-redis
my-redis
PS C:\Windows\system32> docker rm my-redis
my-redis
PS C:\Windows\system32> docker rmi redis
Untagged: redis:latest
Deleted: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
PS C:\Windows\system32>
```

3.CREATING A CUSTOM DOCKER FILE

Creating a new Folder named redis1

Creating a one more new folder inside it with the name of redis to fetch the redis custom image.

Create a new File name Dockerfile without any extensions.



Open the file using any editor example notepad and Copy the Following Code into the file which fetches the redis of latest version and runs the redis server.



Open the Redis folder in a cmd prompt using administrator mode.

Building the image with the current image using the cmd docker build -t redisnew .

```
Administrator: Windows PowerShell
Start a build
PS C:\DockerProjects\Redis> docker build -t redisnew .
[+] Building 0.1s (1/1) FINISHED
=> [internal] load build definition from Dockerfile
=> > transferring dockerfile: 2B
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
PS C:\DockerProjects\Redis> docker build -t redisnew .
[+] Building 158.6s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> > transferring dockerfile: 78B
=> [internal] load metadata for docker.io/library/redis:latest
=> [auth] library/redis:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> > transferring context: 2B
=> [1/1] FROM docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6
=> > resolve docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6a
=> > sha256:9501a6ec095f4bd5242b8ceefdf1119d098b45c6e9237c53cedb022a0ce9c1fa 1.10kB / 1.10kB
=> > sha256:4f4fb700ef54461cf02571ae0db9a0dc1e0cd85577484a6d75e68dc38e8acc1 32B / 32B
=> > sha256:8242f9d5b464f375c010545626f3107bcfe0f68e7a2a78e95c8d397427a12c0e 572B / 572B
=> > sha256:75dffaa679c9b360fd358682b0842b9f32a732a6f80d47e8a7f3efde5f05dc07 1.05MB / 1.44MB
=> > sha256:98e7597530ef912e6a2cce5e6e0cd3f155b611a5d6b6ea7823640b6c4f2c4a30 874B / 874B
=> > sha256:141f00d6fee86bf601ac2eca3db312a24d3e4147bc166bbecb09428be54794b5 97B / 97B
=> > sha256:bc0965b23a04fe7f2d9fb20f597e008fcf89891de1c705ffc1c80483a1f098e4f 28.23MB / 28.23MB
=> > sha256:8912a88e73c83f622d19fd2dac8a055ca96342628d29d56b81fde09475ef7007 15.33MB / 15.33MB
=> exporting to image
=> > exporting layers
=> > exporting manifest sha256:ee573e1f6df4901e37acdc97884314eb51265ccb9c34333ca776a04cf167907a
=> > exporting config sha256:b811c987af0831dd5a1712170ea61a95f065f13a4948c8af664a8b547c4d1928
=> > exporting attestation manifest sha256:31e5120c1d4a80bf8047f43c8e6cb545808ddbcebfe24bd8f4d4343c24ed2eb6
=> > exporting manifest list sha256:9b45b2131bbda720304b9c91c984547b0cc2f050bf13fccd7cb9259ceb65b8f2
=> > naming to docker.io/library/redisnew:latest
=> > unpacking to docker.io/library/redisnew:latest
PS C:\DockerProjects\Redis>
```

Run the command docker images to view all the images available

Running the new latest redis image that we created.

Viewing the docker container that are running.

Stopping the docker container od redislatest.

```
>=> naming to docker.io/library/redisnew:latest
>=> unpacking to docker.io/library/redisnew:latest
$ C:\DockerProjects\Redis> docker run --name myredisnew -d redisnew
42d100567ab6d179e5dd6b79db8056e3d3fe490967e14e7c111476db174da8
$ C:\DockerProjects\Redis> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
42d100567ab redisnew "docker-entrypoint.s..." 8 seconds ago Up 7 seconds 6
myredisnew
c81bd6703d8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 36 minutes ago Up 36 minutes 1
59687->22/tcp, 127.0.0.1:59688->2376/tcp, 127.0.0.1:59690->5000/tcp, 127.0.0.1:59691->8443/tcp, 127.0.0.1:59689
cp minikube
$ C:\DockerProjects\Redis> docker stop myredisnew
myredisnew
$ C:\DockerProjects\Redis> docker login
Authenticating with existing credentials...
Login Succeeded
$ C:\DockerProjects\Redis>
```

Commit the changes into the system using the command docker commit contianerId newname. Now using this new name i.e your username folloew by /name of the container is created in to docker hub

```
PS C:\DockerProjects\Redis> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\DockerProjects\Redis> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED NAMES STATUS PORTS
b42d100567ab redisnew "docker-entrypoint.s..." About a minute ago Exited (0) 42 seconds ago myredisnew
bc81bd6703d8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 37 minutes ago Up 37 minutes 127.0.0.1:59687->22/tcp, 127.0.0.1:59688->237
6/tcp, 127.0.0.1:59690->5000/tcp, 127.0.0.1:59691->8443/tcp, 127.0.0.1:59689
PS C:\DockerProjects\Redis> docker commit 0e993d2009a1 laksh/redis1
Error response from daemon: No such container: 0e993d2009a1
PS C:\DockerProjects\Redis> docker commit b42d100567ab laksh/redis1
sha256:03597bf9041c90825e103b9797bd7c63bf0b05402af01b11a0bd27c53063776bf
PS C:\DockerProjects\Redis> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
laksh/redis1 latest 03597bf9041c 18 seconds ago 173MB
docker/desktop-kubernetes kubernetes-v1.30.5-cni-v1.4.0-critools-v1.29.0-cri-docker-v0.3.11-1-debian 7a7b02256c8d 7 weeks ago 625MB
redisnew latest 9b45b2131bb0 2 months ago 173MB
registry.k8s.io/kube-apiserver v1.30.5 7746ea55ad74 2 months ago 153MB
registry.k8s.io/kube-controller-manager v1.30.5 bbd15d267294 2 months ago 146MB
registry.k8s.io/kube-scheduler v1.30.5 62c91756a3c9 2 months ago 84.6MB
registry.k8s.io/kube-proxy v1.30.5 fa20f91153b9 2 months ago 118MB
gcr.io/k8s-minikube/kicbase v0.0.45 81df28859520 3 months ago 1.81GB
<none> <none> e7c9b3bc515 3 months ago 1.81GB
registry.k8s.io/coredns/coredns v1.11.3 9caabbff6238b 4 months ago 85.1MB
registry.k8s.io/etcd 3.5.12-0 44a8e24dcdbba 10 months ago 211MB
docker/desktop-vpnkit-controller dc331cb22850be0cd97c84a9cfecaf44a1afb6e 7ecf567ea070 19 months ago 47MB
registry.k8s.io/pause 3.9 7031c1b28338 2 years ago 1.07MB
docker/desktop-storage-provisioner v2.0 115d77efee6e2 3 years ago 59.2MB
registry.k8s.io/etcd 3.5.15-0 a6dc63e6e8cf 54 years ago 56.9MB
PS C:\DockerProjects\Redis>
```

We can see that the images is saved locally by the name iamabdurrasheed/redisnew.

Using the docker push nameOfimage we can push our hub into the docke hub cloud.

```
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker push iamabdurrasheed/redisnew
Using default tag: latest
The push refers to repository [docker.io/iamabdurrasheed/redisnew]
74c736b00471: Pushed
8fb865c3d417: Pushed
928f5dbb5007: Pushed
4f4fb700ef54: Pushed
2d429b9e73a6: Pushed
92ef1eccbb9f: Pushed
6fd0c1bf3b91: Pushed
5e00ad97561c: Pushed
latest: digest: sha256:81df38b111bc2223b89aedf3fd238fa7f206e067a2eb0e95c0b146d6160efa5f size: 2038
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

Image after it is pushed into the docker hub.

Docker Desktop Build history:

Name	ID	Builder	Duration	Created	Author
Redis	afedrm	default	2m 06s	22 hours ago	N/A
Redis	e11kcu	default	0.0s	22 hours ago	N/A
Redis	q7t85b	default	0.3s	22 hours ago	N/A

Docker Hub Repository List:

Name	Last Pushed	Contains	Visibility	Scout
luxxx000/redis1	about 22 hours ago		Public	Inactive
luxxx000/laksh	about 22 hours ago		Public	Inactive

Now removing the images and container locally.

Pulling the image from the hub that we just pushed into the docker hub.

Running the new redis file just we just pulled.

Opening the interactive redis cli and setting the name and getting the name that we stored temporarily.

Stopping the container that was running.

Deleting the container and images after it is stopped.

```

Administrator: Windows PowerShell
e7c9bc3bc515 3 months ago 1.81GB
gcr.io/k8s-minikube/kicbase
81df28859520 3 months ago 1.81GB
registry.k8s.io/coredns/coredns
9caabbff6238b 4 months ago 85.1MB
registry.k8s.io/etcd
44a8e24dcbb4 10 months ago 211MB
docker/desktop-vpnkit-controller
7ecf567ea070 19 months ago 47MB
registry.k8s.io/pause
7031c1b28338 2 years ago 1.07MB
docker/desktop-storage-provisioner
115d77fe6e2 3 years ago 59.2MB
registry.k8s.io/etcd
a6dc63e6e8cf 54 years ago 56.9MB
PS C:\DockerProjects\Redis> docker run --name myredisnew -d redisnew
docker: Error response from daemon: Conflict. The container name "/myredisnew" is already in use by container "b42d100567ab"
b42d100567ab
PS C:\DockerProjects\Redis> docker stop b42d100567ab
b42d100567ab
PS C:\DockerProjects\Redis>

```

3.MODIFY AND PUSH :DOCKER, UBUNTU AND GIT

1. Login to the docker using docker login -u username. In administration mode
2. Let us first pull the ubuntu image . Pulling the latest ubuntu image.
3. Running the ubuntu image in a container that we just pulled.
4. Now let us enter into ubuntu interactive operation of git.
5. Updating the apt using the command “apt update”.

```
root@bc654e86c919:/  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Windows\system32> docker login  
Authenticating with existing credentials...  
Login Succeeded  
PS C:\Windows\system32> docker pull ubuntu  
Using default tag: latest  
latest: Pulling from library/ubuntu  
de44b265507a: Download complete  
Digest: sha256:89dd3c3b9c6cecbf1667e9290b3bc61b78c2c2bd8e5f0fea92cc6734ab  
Status: Downloaded newer image for ubuntu:latest  
docker.io/library/ubuntu:latest  
PS C:\Windows\system32> docker run -it --name newubuntu -d ubuntu  
bc654e86c919bbbed567116b82d6714c7bec831421a56b4183306f00ff0ef9f04  
PS C:\Windows\system32> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
bc654e86c919 ubuntu "/bin/bash" 5 seconds ago Up 4 seconds newubuntu  
PS C:\Windows\system32> docker exec -it bc654e86c919 bash  
root@bc654e86c919:~# git --version  
bash: git: command not found  
root@bc654e86c919:~# apt update  
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]  
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [627 kB]  
Get:4 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [726 kB]  
Get:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:6 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [15.3 kB]  
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [607 kB]  
Get:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:9 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]  
Get:10 http://archive.ubuntu.com/ubuntu/noble/universe amd64 Packages [19.3 MB]  
Get:11 http://archive.ubuntu.com/ubuntu/noble/main amd64 Packages [1808 kB]  
Get:12 http://archive.ubuntu.com/ubuntu/noble/restricted amd64 Packages [117 kB]  
Get:13 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 Packages [858 kB]  
Get:14 http://archive.ubuntu.com/ubuntu/noble-updates/universe amd64 Packages [938 kB]  
Get:15 http://archive.ubuntu.com/ubuntu/noble-updates/multiverse amd64 Packages [19.7 kB]  
Get:16 http://archive.ubuntu.com/ubuntu/noble-updates/restricted amd64 Packages [628 kB]  
Get:17 http://archive.ubuntu.com/ubuntu/noble-backports/universe amd64 Packages [11.9 kB]  
Fetched 26.6 MB in 29s (904 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
root@bc654e86c919:~#
```

6. Installing the git using apt install git -y.

```
root@bc654e86c919:/  
root@bc654e86c919:~# apt install git -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  adduser ca-certificates git-man krb5-locales less liblbbtli1 libbbsd0 liblbcior0.10 libcurl3t64-grnutils libedit2 liberror-perl libexpat1 libfido2-1 libgdbm-compat4t64  
  libgdbm6t64 libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 libnghtp2-14 libperl5.38t64 libpsl5t64 librmp1 libssasl2-2  
  libssasl2-modules libssasl2-modules-db libssh-4 libx11-6 libx11-data libxauf0 libxcb1 libxdmcp6 libxext6 libxmuu1 netbase openssh-client openssl patch perl  
  perl-modules-5.38 publicsuffix xauth  
Suggested packages:  
  liblocale-gettext-perl cron quota encrypts-utils gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn  
  gdm-110n krb5-doc krb5-user libssasl2-modules-gssapi-mit | libssasl2-modules-gssapi-heimdal libssasl2-modules-ldap libssasl2-modules-otp libssasl2-modules-sql keychain  
  libpam-sha512 cryptopk ssh-askpass ed diffutils-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl  
The following NEW packages will be installed:  
  adduser ca-certificates git-man krb5-locales less liblbbtli1 libbbsd0 liblbcior0.10 libcurl3t64-grnutils libedit2 liberror-perl libexpat1 libfido2-1 libgdbm-compat4t64  
  libgdbm6t64 libgssapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 libnghtp2-14 libperl5.38t64 libpsl5t64 librmp1 libssasl2-2  
  libssasl2-modules libssasl2-modules-db libssh-4 libx11-6 libx11-data libxauf0 libxcb1 libxdmcp6 libxext6 libxmuu1 netbase openssh-client openssl patch perl  
  perl-modules-5.38 publicsuffix xauth  
0 upgraded, 46 newly installed, 0 to remove and 0 not upgraded.  
Need to get 18.9 MB of archives.  
After this operation, 92.7 MB of additional disk space will be used.  
Get:1 http://archive.ubuntu.com/ubuntu/noble/main amd64 perl-modules-5.38 all 5.38.2-3.2build2 [3118 kB]  
Get:2 http://archive.ubuntu.com/ubuntu/noble/main amd64 libgdbm6t64 amd64 1.23-5.1build1 [34.4 kB]  
Get:3 http://archive.ubuntu.com/ubuntu/noble/main amd64 libgdbm-compat4t64 amd64 1.23-5.1build1 [6710 B]  
Get:4 http://archive.ubuntu.com/ubuntu/noble/main amd64 libperl5.38t64 amd64 5.38.2-3.2build2 [4873 kB]  
Get:5 http://archive.ubuntu.com/ubuntu/noble/main amd64 perl amd64 5.38.2-3.2build2 [231 kB]  
Get:6 http://archive.ubuntu.com/ubuntu/noble/main amd64 adduser all 3.137ubuntu1 [101 kB]  
Get:7 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 openssl 3.0.13-ubuntus3.4 [1003 kB]  
Get:8 http://archive.ubuntu.com/ubuntu/noble/ca-certificates all 20240203 [159 kB]  
Get:9 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 krb5-locales all 1.20.1-6ubuntu2.2 [14.0 kB]  
Get:10 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 less amd64 590-2ubuntu0.1 [142 kB]  
Get:11 http://archive.ubuntu.com/ubuntu/noble/main amd64 libbbsd0 amd64 0.12.1-1build1 [41.2 kB]  
Get:12 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.1 [87.1 kB]  
Get:13 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 libkrb5support0 amd64 1.20.1-6ubuntu2.2 [33.7 kB]  
Get:14 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 libk5crypto3 amd64 1.20.1-6ubuntu2.2 [81.8 kB]  
Get:15 http://archive.ubuntu.com/ubuntu/noble/main amd64 libkeyutils1 amd64 1.6.3-3build1 [5498 B]  
Get:16 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 libkrb5-3 amd64 1.20.1-6ubuntu2.2 [347 kB]  
Get:17 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 libgssapi-krb5-2 amd64 1.20.1-6ubuntu2.2 [143 kB]
```

Git installation.

7. Checking the git version using git –version.
8. Exiting the git using exit command.

9. Stopping the git container that we just started
10. Commit the changes and saving the image.
11. Checking the docker images.

```
Administrator: Windows PowerShell
root@cc654e86c919:/# git --version
git version 2.43.0
root@cc654e86c919:/# exit
exit
PS C:\Windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
bc654e86c919 ubuntu "/bin/bash" 4 minutes ago Up 4 minutes newubuntu
PS C:\Windows\system32> docker stop bc654e86c919
bc654e86c919
PS C:\Windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Windows\system32> docker commit bc654e86c919 luxxx000/newubuntu
sha256:5ca211b545fac6029754a5ce8db986204b41adee76ec6eb1d13e33751574d26
PS C:\Windows\system32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
luxxx000/newubuntu latest 5ce211b545fa 17 seconds ago 310MB
simple-calculator latest 3e5a2ca9a2d3 18 minutes ago 171MB
luxxx000/simple-calculator latest 3e5a2ca9a2d3 18 minutes ago 171MB
laksh/redis1 latest 035970f9041c 24 hours ago 173MB
wordpress latest 2f3572d5cd72 2 weeks ago 993MB
ubuntu latest 80dd3c3b9c6c 2 weeks ago 117MB
docken/desktop-kubernetes kubernetes-v1.30.5-cni-v1.4.0-critools-v1.29.0-cri-dockerd-v0.3.11-1-debian latest 7a7b02256c8d 7 weeks ago 625MB
redisnew latest 9b45b2131b1bd 2 months ago 173MB
registry.k8s.io/kube-apiserver v1.30.5 7746ea55ad74 2 months ago 153MB
registry.k8s.io/kube-scheduler v1.30.5 62c91756a3c9 2 months ago 84.6MB
registry.k8s.io/kube-controller-manager v1.30.5 bbd15d267294 2 months ago 146MB
registry.k8s.io/kube-proxy v0.0.45 fa20f91153b9 2 months ago 118MB
gcr.io/k8s-minikube/kibbase <none> 81df28859520 3 months ago 1.81GB
<none> <none> e7c9bc3bc515 3 months ago 1.81GB
registry.k8s.io/coredns/coredns v1.11.3 9caaabbff6238b 4 months ago 85.1MB
registry.k8s.io/etcfd 3.5.12-0 44a824dcbbfa 10 months ago 211MB
mysql 5.7 4bc6bc963e6d 12 months ago 689MB
phpmyadmin/phpmyadmin latest 67ba2559fd00 16 months ago 803MB
docken/desktop-vpnkit-controller dc331cb22850be0cdd97c84a9cfecaf44a1afb6e latest 7ecf567ea070 19 months ago 47MB
registry.k8s.io/pause 3.9 7031c1b28338 2 years ago 1.07MB
docken/desktop-storage-provisioner v2.0 115d77fe6e62 3 years ago 59.2MB
registry.k8s.io/etcfd 3.5.15-0 a6dc63e6e8ecf 54 years ago 56.9MB
PS C:\Windows\system32>
```

12. Login to the docker hub using docker login.
13. Push the latest updated image into the docker hub.

```
Administrator: Windows PowerShell
registry.k8s.io/etcfd 3.5.15-0
PS C:\Windows\system32> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Windows\system32> docker push luxxx000/newubuntu
Using default tag: latest
The push refers to repository [docker.io/luxxx000/newubuntu]
ba575d28d80a: Pushed
de44b265597a: Mounted from library/ubuntu
latest: digest: sha256:5ca211b545fac6029754a5ce8db986204b41adee76ec6eb1d13e33751574d26 size: 751
PS C:\Windows\system32>
```

14. We can see that the ubuntu image is uploaded into the docker hub.

The screenshot shows the Docker Hub interface. At the top, there's a search bar with 'hub.docker.com' and a navigation bar with 'Explore', 'Repositories', 'Organizations', and 'Usage'. Below the search bar, there's a dropdown menu set to 'luxxx000' and a general search bar. To the right of these are buttons for 'Create a repository', 'Search Docker Hub', and other account-related icons. The main content area displays a table of repositories under the heading 'luxxx000'. The table columns are 'Name', 'Last Pushed', 'Contains', 'Visibility', and 'Scout'. The listed repositories are:

- luxxx000/newubuntu (Last pushed 1 minute ago, IMAGE, Public, Inactive)
- luxxx000/simple-calculator (Last pushed 17 minutes ago, IMAGE, Public, Inactive)
- luxxx000/redis1 (Last pushed 1 day ago, IMAGE, Public, Inactive)
- luxxx000/laksh (Last pushed 1 day ago, IMAGE, Public, Inactive)

 At the bottom of the table, it says '1-4 of 4'. To the right of the table, there's a promotional section for organizations with a 'Create an organization' button and some icons.

15. Deleting the container locally .

16. Deleting the image locally.
17. We can see the container is deleting.
18. Pulling the image that we just pushed.

```

Administrator: Windows PowerShell
Login Succeeded
PS C:\Windows\system32> docker push luxxx000/newubuntu
Using default tag: latest
The push refers to repository [docker.io/luxxx000/newubuntu]
ba575d28d80a: Pushed
de44b265597a: Mounted from library/ubuntu
latest: digest: sha256:5ce21b545fac6029754a5ce8db986204b41adee76ec6eb11d13e33751574d26 size: 751
PS C:\Windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
bc654e86c919 ubuntu "/bin/bash" 10 minutes ago Exited (137) 5 minutes ago
newubuntu
83d4c8d33678 luxxx000/simple-calculator "docker-entrypoint.s..." 14 minutes ago Exited (0) 14 minutes ago
quizzical_germain
bc1bd703d8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 25 hours ago Exited (255) 3 hours ago 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/t
cp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\Windows\system32> docker rm bc654e86c919
bc654e86c919
PS C:\Windows\system32> docker rmi luxxx000/newubuntu:latest
Untagged: luxxx000/newubuntu:latest
Deleted: sha256:5ce21b545fac6029754a5ce8db986204b41adee76ec6eb11d13e33751574d26
PS C:\Windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED NAMES STATUS PORTS
83d4c8d33678 luxxx000/simple-calculator "docker-entrypoint.s..." 15 minutes ago Exited (0) 15 minutes ago
quizzical_germain
bc1bd703d8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 25 hours ago Exited (255) 3 hours ago 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/t
cp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
Using default tag: latest
latest: Pulling from luxxx000/newubuntu
ba575d28d80a: Download complete
Digest: sha256:5ce21b545fac6029754a5ce8db986204b41adee76ec6eb11d13e33751574d26
Status: Downloaded newer image for luxxx000/newubuntu:latest
docker.io/luxxx000/newubuntu:latest
PS C:\Windows\system32>

```

19. Running the image into a container.
20. Checking the git version and making sure that the same git version is seen.

```

root@bc654e86c919:~/
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Windows\system32> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Windows\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
de44b265597a: Download complete
Digest: sha256:80dd3c3b9c6ecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Windows\system32> docker run -it --name newubuntu -d ubuntu
bc654e86c919:82d6714c7bec8314213a56b4183306f00f0e9f04
PS C:\Windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
bc654e86c919 ubuntu "/bin/bash" 5 seconds ago Up 4 seconds newubuntu
PS C:\Windows\system32> docker exec -it bc654e86c919 bash
root@bc654e86c919:/# git --version
bash: git: command not found
root@bc654e86c919:/# 

```

21. Exiting the Git.
22. Deleting the container that we pulled from docker hub locally.
23. Deleting the image that we pulled from docker hub locally

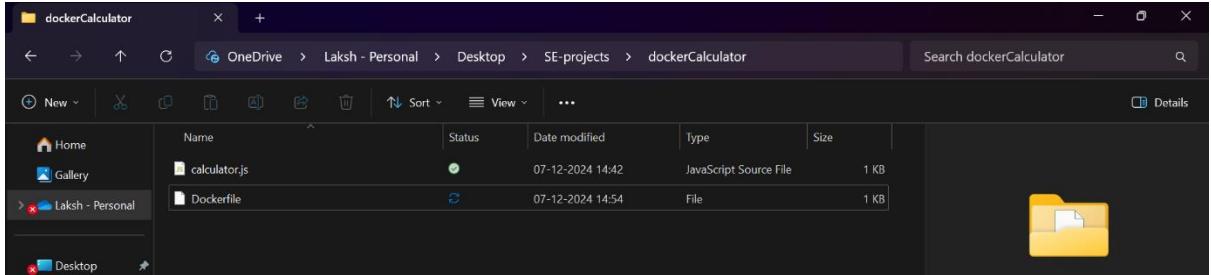
```

Administrator: Windows PowerShell
PS C:\Windows\system32> docker pull luxxx000/newubuntu
Using default tag: latest
latest: Pulling from luxxx000/newubuntu
ba575d28d80a: Download complete
Digest: sha256:5ce21b545fac6029754a5ce8db986204b41adee76ec6eb11d13e33751574d26
Status: Downloaded newer image for luxxx000/newubuntu:latest
PS C:\Windows\system32> docker run --name newubuntu -it luxxx000/newubuntu
root@6d66b526a3a9:/# git --version
git version 2.45.0
root@6d66b526a3a9:/# exit
exit
PS C:\Windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED NAMES STATUS PORTS
6d66b526a3a9 luxxx000/newubuntu "/bin/bash" 34 seconds ago Exited (0) 10 seconds ago
newubuntu
83d4c8d33678 luxxx000/simple-calculator "docker-entrypoint.s..." 20 minutes ago Exited (0) 20 minutes ago
quizzical_germain
bc1bd703d8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 25 hours ago Exited (255) 3 hours ago 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/t
cp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\Windows\system32> docker rm 6d66b526a3a9
6d66b526a3a9
PS C:\Windows\system32> docker rmi luxxx000/newubuntu
Untagged: luxxx000/newubuntu:latest
Deleted: sha256:5ce21b545fac6029754a5ce8db986204b41adee76ec6eb11d13e33751574d26
PS C:\Windows\system32>

```

4.CREATE AND PUSH DOCKER FILE IMAGE

Create a Folder named dockerCalculator.



Using command prompt create a file with no extension

Inside that Folder create 2 files DockerFile and Calulator.js file with the Js calculator program.

DockerFile and Calulator.js file..

```
calculator.js
1 // calculator.js
2 // Function to add two numbers
3 function add(a, b) {
4     return a + b;
5 }
6 // Function to subtract two numbers
7 function subtract(a, b) {
8     return a - b;
9 }
10 // Function to multiply two numbers
11 function multiply(a, b) {
12     return a * b;
13 }
14 // Function to divide two numbers
15 function divide(a, b) {
16     if (b === 0) {
17         return "Cannot divide by zero!";
18     }
19     return a / b;
20 }
21 // Print the calculations
22 console.log("Addition (2 + 3):", add(2, 3));
23 console.log("Subtraction (5 - 2):", subtract(5, 2));
24 console.log("Multiplication (4 * 3):", multiply(4, 3));
25 console.log("Division (10 / 2):", divide(10, 2));
```

```
Dockerfile
1 FROM node:16-alpine
2 WORKDIR /app
3 COPY calculator.js /app
4 CMD ["node", "calculator.js"]
```

Now Let us Navigate to that path and build the image using the following command.Image of Calculator is Build successfully.

```

PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker build -t simple-calculator .
[+] Building 0.2s (1/1) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 45B
ERROR: failed to solve: Internal: Internal: Internal: stream terminated by RST_STREAM with error code: INTERNAL_ERROR
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/a4i0ovxqqlj8zxblickfjeelv
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker build -t simple-calculator
[+] Building 90.6s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 128B
=> [internal] load metadata for docker.io/library/node:16-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerrcignore
=> transferring context: 2B
=> [1/3] FROM docker.io/library/node:16-alpine@sha256:a1f9d027912b58a7c75be7716c97cfbc6d3099f3a97ed84aa490be9dee
=> resolve docker.io/library/node@sha256:a1f9d027912b58a7c75be7716c97cfbc6d3099f3a97ed84aa490be9dee
=> sha256:d9059661ce70092af66d2773666584fc8addcb78a2be63f720022f4875577ea9 452B / 452B
=> sha256:93b3025fe10392717d06ec0d012a9ffa2039d766a322aac899c6831dd93382c2 2.34MB / 2.34MB
=> sha256:eee371b9ce3ffdbb8a703b9a14d318801ddc3468f096bb6cfeabeb715147f9 36.63MB / 36.63MB
=> sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 3.40MB / 3.40MB
=> extracting sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de
=> extracting sha256:eee371b9ce3ffdbb8a703b9a14d318801ddc3468f096bb6cfeabeb715147f9
=> extracting sha256:93b3025fe10392717d06ec0d012a9ffa2039d766a322aac899c6831dd93382c2
=> extracting sha256:d9059661ce70092af66d2773666584fc8addcb78a2be63f720022f4875577ea9
=> [internal] load build context
=> transferring context: 694B
=> [2/3] WORKDIR /app
=> [3/3] COPY calculator.js /app
=> exporting to image
=> exporting layers
=> exporting manifest sha256:2b16783d38407e30a2484f968cfdf863d51a6bf103ab3a0a9c040db3b285ee9
=> exporting config sha256:92a9ae7c8f81b1c765c005e2d8d548e7348c7673f520e6b60284ec9d2fb36de
=> exporting attestation manifest sha256:e1f0e20d94c348c18264454cfed9f2db517f12c5522191d3e579f3a0199905ef
=> exporting manifest list sha256:3e5ac2a9a2d3c8ad584ea189551bb6f2a5baeaf8f7fb717793b867263b290a66
=> naming to docker.io/library/simple-calculator:latest
=> unpacking to docker.io/library/simple-calculator:latest

```

Running the Calculator image.

Login to docker hub to push the image.

Using docker tag command to save the changes into the image.

Pushing the image into my docker hub .

```

PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker run simple-calculator
Addition (2 + 3): 5
Subtraction (5 - 2): 3
Multiplication (4 * 3): 12
Division (10 / 2): 5
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker tag simple-calculator luxxx000/simple-calculator
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker push luxxx000/simple-calculator
Using default tag: latest
The push refers to repository [docker.io/luxxx000/simple-calculator]
eee371b9ce3f: Pushed
93b3025fe103: Pushed
7264a8db6415: Pushed
d9059661ce70: Pushed
9366b21e9a44: Pushed
aafdda26893d: Pushed
535e5c35d92e: Pushed
latest: digest: sha256:3e5ac2a9a2d3c8ad584ea189551bb6f2a5baeaf8f7fb717793b867263b290a66 size: 856
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> |

```

We can see that image is pushed into the docker hub.

Removing the container locally.

Removing the image locally

Pulling the latest image that we just pushed into the docker hub.

Running the image that we just pulled from docker hub.

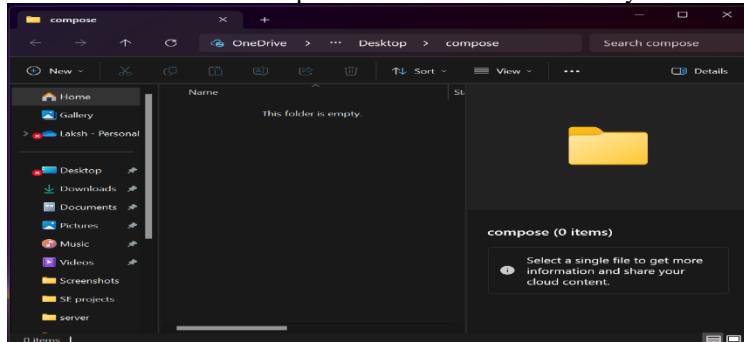
```
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
27e0751fab69 simple-calculator "docker-entrypoint.s..." 5 minutes ago Exited (0) 5 minutes ago
eloquent_f
eistel
bc81bd6703d8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 24 hours ago Exited (255) 2 hours ago 127.0.0.1:32
768->22/tcp, 127.0.0.1:32769->2376/tcp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker rm 27e0751fab69
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker rmi luxxx000/simple-calculator
Untagged: luxxx000/simple-calculator:latest
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker pull luxxx000/simple-calculator
Using default tag: latest
latest: Pulling from luxxx000/simple-calculator
Digest: sha256:3e5ac2a9a2d3c8ad584ea189551bb6f2a5baeaf8f7fb717793b867263b290a66
Status: Downloaded newer image for luxxx000/simple-calculator:latest
docker.io/luxxx000/simple-calculator:latest
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\dockerCalculator> docker run luxxx000/simple-calculator
Addition (2 + 3): 5
Subtraction (5 - 2): 3
Multiplication (4 * 3): 12
```

5.DOCKER COMPOSE

RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE

Login to docker using command prompt.

Create a folder named compose in which we will create a yaml file.



Go to <https://gist.github.com/bradtraversy/faa8de544c62eef3f31de406982f1d42> to copy the yml file for running mysql and wordpress in a single container.

A screenshot of a GitHub Gist page. The title is 'Docker Compose File For Wordpress, MySQL & phpmyadmin' by bradtraversy. It has 801 stars and 398 forks. The code editor shows a YAML file named 'docker_wordpress.md' with the following content:

```
version: '3'

services:
  # Database
  db:
    image: mysql:5.7
    volumes:
```

Now create a docker-compose.yaml file and paste the content that we copied.

```

version: '3'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
    networks:
      - wpsite
  phpmyadmin:
    depends_on:
      - db
    image: phpmyadmin/phpmyadmin
    restart: always
    ports:
      - '8080:80'
    environment:
      PMA_HOST: db
      MYSQL_ROOT_PASSWORD: password
    networks:
      - wpsite
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:

```

Navigate to the folder that we just created.

Check if the docker-compose Is available by checking its version using the following command.

```

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

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

PS C:\Windows\system32> cd C:\Users\laksh\OneDrive\Desktop\compose
PS C:\Users\laksh\OneDrive\Desktop\compose> docker compose --version

Usage: docker compose [OPTIONS] COMMAND

Define and run multi-container applications with Docker

Options:
  --all-resources           Include all resources, even those not
                           used by services
  --ansi string             Control when to print ANSI control
                           characters ("never"|"always"|"auto")
                           (default "auto")
  --compatibility          Run compose in backward compatibility mode
  --dry-run                 Execute command in dry run mode
  --env-file stringArray    Specify an alternate environment file
  -f, --file stringArray    Compose configuration files
  --parallel int            Control max parallelism, -1 for
                           unlimited (default -1)
  --profile stringArray     Specify a profile to enable
  --progress string          Set type of progress output (auto,
                           tty, plain, json, quiet) (default "auto")
  --project-directory string Specify an alternate working directory
                           (default: the path of the, first
                           specified, Compose file)

```

Run docker-compose up -d that will pull the images and run them in a single container ..We can see that we the mysql and wordpress are running in the single container

```

Administrator: Windows PowerShell
Recy wait      Block until containers of all (or specified) services stop.
watch         Watch build context for service and rebuild/refresh containers when files are updated

Run 'docker compose COMMAND --help' for more information on a command.
PS C:\Users\laksh\OneDrive\Desktop\compose> docker-compose up -d
time="2024-12-07T14:23:45+05:30" level=warning msg="C:\\Users\\laksh\\OneDrive\\Desktop\\compose\\docker-compose.yaml: t
he attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 1/21
- wordpress [       ] Pulling
- db [        ] Pulling
- phpmyadmin [      ] Pulling
               6.2s
               6.2s
               6.2s

```

Check if the container is created using docker ps -a.

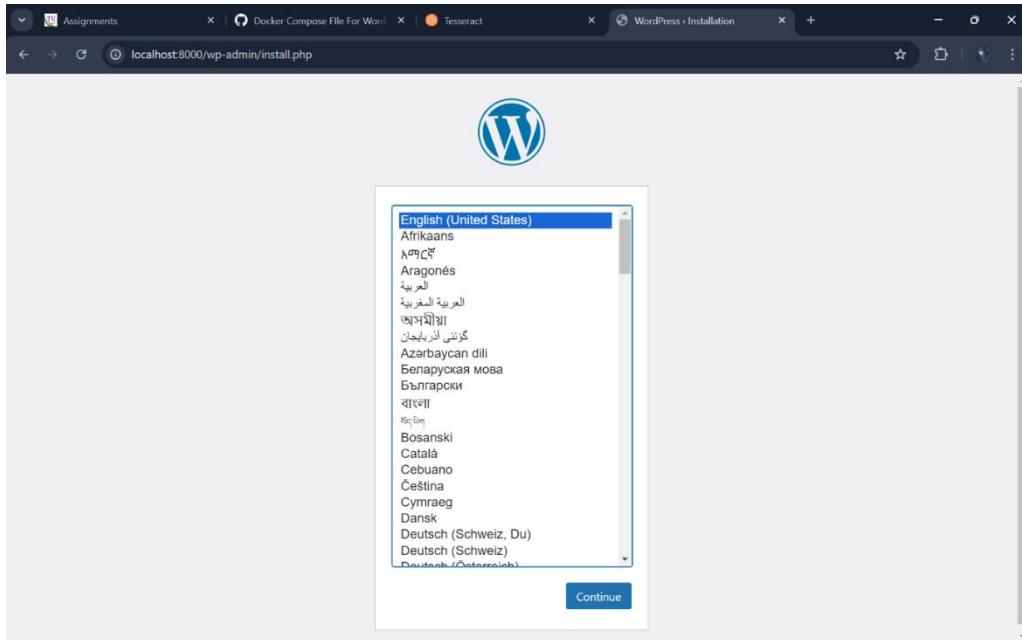
```

Administrator: Windows PowerShell
$ cd, please remove it to avoid potential confusion"
[+] Running 5/1
  ✓ db Pulled
  ✓ wordpress Pulled
  ✓ phpmadmin Pulled

[+] Running 4/5
  ✓ Network compose_wpsite      Created
  ✓ Volume "compose_db_data"    Created
  ✓ Container compose-db-1     Started
  ✓ Container compose-wordpress-1 Started
    Container compose-phpmyadmin-1 Starting
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 0.0.0.0:0: listen tcp 0.0.0.0:8080: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
PS C:\Users\laksh\OneDrive\Desktop\compose> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED NAMES STATUS PORTS
fc640e02f2ed wordpress.latest      "docker-entrypoint.s..." 34 seconds ago Up 33 seconds      0.0.0.0:8080->80/tcp
16d15b540134 phpmadmin/phpmyadmin   "/docker-entrypoint.s..." 34 seconds ago Created          compose-phpmyadmin-1
415e8ca26c68 mysql:5.7            "docker-entrypoint.s..." 35 seconds ago Up 33 seconds      3306/tcp, 33060/tcp
bc817d6783d8 gcr.io/k8s-minikube/kicbase:v0.45  "/usr/local/bin/entr..." 24 hours ago Exited (255) 2 hours ago  127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp
, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\Users\laksh\OneDrive\Desktop\compose>

```

Now since the container is running visit the localhost:8000 to open the wordpress file.



Filling the required information to Register into wordporess.

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title dockerse

Username luxxx000

Usetnames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password Show Medium

Your Email lakshvijay04@gmail.com

Double-check your email address before continuing.

Search engine visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

[Install WordPress](#)

Continue to login.

Success!

WordPress has been installed. Thank you, and enjoy!

Username luxxx000

Password Your chosen password.

[Log In](#)

Login with the credentials that you just provided.

Log In — dockerse — WordPress

Username or Email Address luxxx000

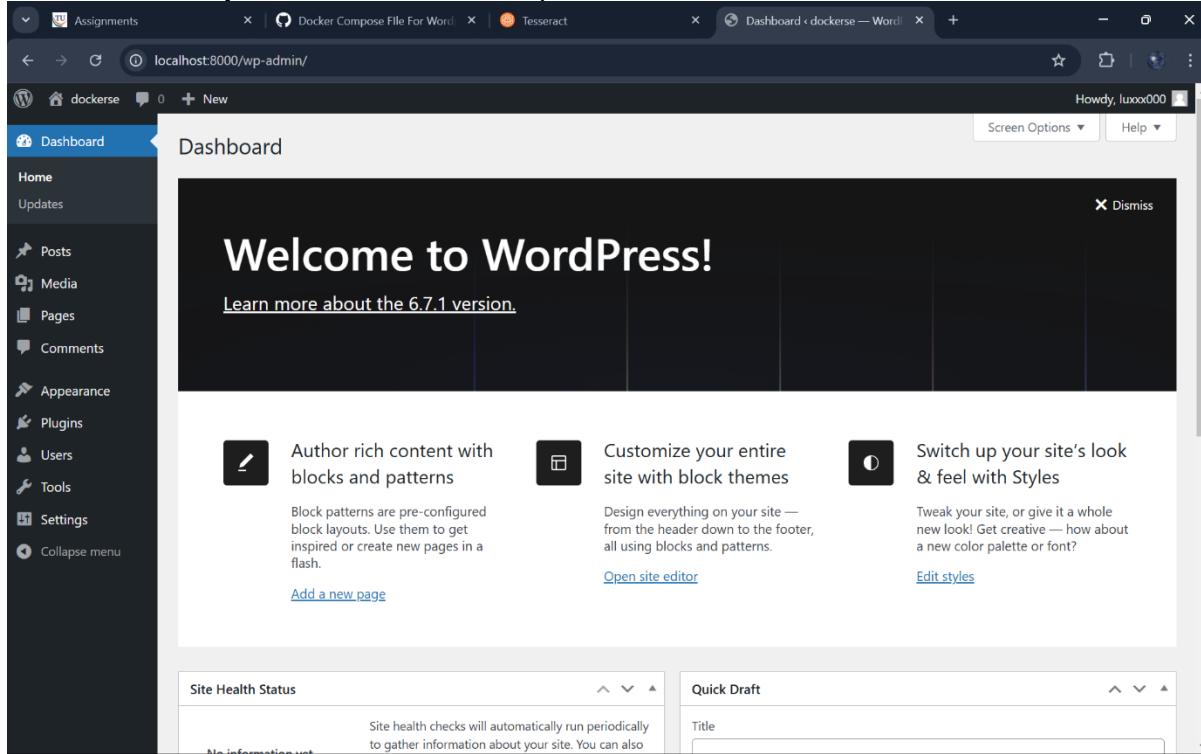
Password

Remember Me [Log In](#)

[Lost your password?](#)

[← Go to dockerse](#)

We have successfully created the account in wordpress inside the docker.



Stop the compose using docker-compose stop.

Starting the compose using docker-compose start.

Stop the compose using docker-compose stop.

Deleting/Removing the compose using docker-compose down.

```
Administrator: Windows PowerShell
PS C:\Users\laksh\OneDrive\Desktop\compose> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
fc48e82faed        wordpress:latest     "docker-entrypoint.s..."   34 seconds ago    Up 33 seconds      0.0.0.0:8000->80/tcp
16d15b540134       phpmyadmin/phpmyadmin   "/docker-entrypoint.s..."   34 seconds ago    Created            compose-phpmyadmin-1
416e8ca26c68       mysql:5.7          "docker-entrypoint.s..."   35 seconds ago    Up 33 seconds      3306/tcp, 33060/tcp
bc81bd6703d8       gcr.io/k8s-minikube/kicbase:v0.0.45  "/usr/local/bin/entr..."  24 hours ago     Exited (255) 2 hours ago  127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp
PS C:\Users\laksh\OneDrive\Desktop\compose> docker-compose stop
time="2024-12-07T14:38:46+05:30" level=warning msg="C:\\Users\\laksh\\OneDrive\\Desktop\\compose\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Stopping 3/3
  ✓ Container compose-wordpress-1 Stopped
  ✓ Container compose-phpmyadmin-1 Stopped
  ✓ Container compose-db-1 Stopped
PS C:\Users\laksh\OneDrive\Desktop\compose> docker-compose start
time="2024-12-07T14:39:01+05:30" level=warning msg="C:\\Users\\laksh\\OneDrive\\Desktop\\compose\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 2/3
  ✓ Container compose-db-1 Started
  ✓ Container compose-phpmyadmin-1 Starting
  ✓ Container compose-wordpress-1 Started
Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:8000 -> 0.0.0.0:0: listen tcp 0.0.0.0:8000: bind: Only one usage of each socket address (proto/endpoint/network address/port) is normally permitted.
PS C:\Users\laksh\OneDrive\Desktop\compose> docker-compose stop
time="2024-12-07T14:39:15+05:30" level=warning msg="C:\\Users\\laksh\\OneDrive\\Desktop\\compose\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Stopping 3/3
  ✓ Container compose-wordpress-1 Stopped
  ✓ Container compose-phpmyadmin-1 Stopped
  ✓ Container compose-db-1 Stopped
PS C:\Users\laksh\OneDrive\Desktop\compose> docker-compose down
time="2024-12-07T14:39:38+05:30" level=warning msg="C:\\Users\\laksh\\OneDrive\\Desktop\\compose\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Stopping 4/4
  ✓ Container compose-phpmyadmin-1 Removed
  ✓ Container compose-wordpress-1 Removed
  ✓ Container compose-db-1 Removed
  ✓ Network compose_wsite Removed
PS C:\Users\laksh\OneDrive\Desktop\compose>
```

6. DEPLOYING AND SCALING APPLICATIONS USING MINIKUBE

Setting Up Minikube and Kubernetes

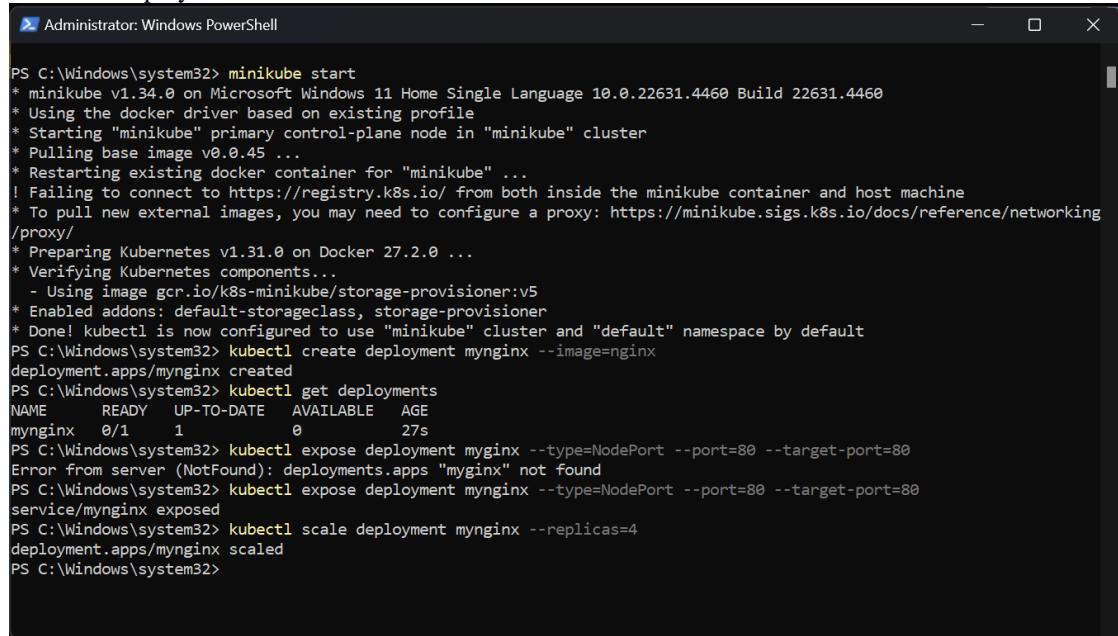
Start minikube in terminal

Create the Nginx Deployment

You can verify if the Nginx deployment was created correctly by running this command

Expose the Service

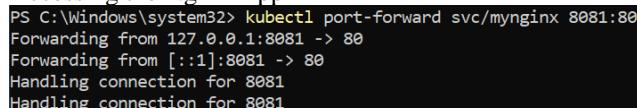
Scale the Deployment



```
Administrator: Windows PowerShell

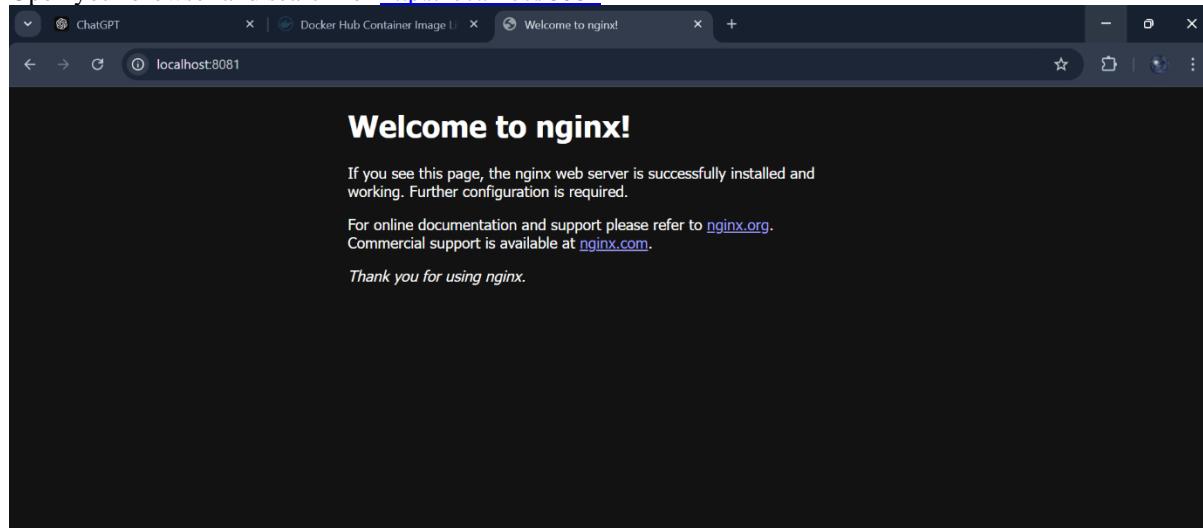
PS C:\Windows\system32> minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4460 Build 22631.4460
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from both inside the minikube container and host machine
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\Windows\system32> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   0/1     1           0           27s
PS C:\Windows\system32> kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80
Error from server (NotFound): deployments.apps "mynginx" not found
PS C:\Windows\system32> kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80
service/mynginx exposed
PS C:\Windows\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Windows\system32>
```

Accessing the Nginx App



```
PS C:\Windows\system32> kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Handling connection for 8081
Handling connection for 8081
```

Open your browser and search for <http://localhost:8081>



Stop the Nginx Deployment and Service

If you're done with Minikube, stop it to free up system resources

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\Windows\system32> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   0/1     1           0           27s
PS C:\Windows\system32> kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80
Error from server (NotFound): deployments.apps "mynginx" not found
PS C:\Windows\system32> kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80
service/mynginx exposed
PS C:\Windows\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Windows\system32> kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Handling connection for 8081
Handling connection for 8081
PS C:\Windows\system32> kubectl delete deployment mynginx
deployment.apps "mynginx" deleted
PS C:\Windows\system32> kubectl delete service mynginx
service "mynginx" deleted
PS C:\Windows\system32> minikube stop
* Stopping node "minikube" ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
PS C:\Windows\system32>
```

7.NAGIOS

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

PS C:\Windows\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
d3245570f968: Pulling fs layer
8c389e58e867: Pulling fs layer
#8aeb2cf4e43: Download complete
#eb77e6dde3e: Download complete
#738fc7528889: Download complete
15f36ddbb439: Download complete
#f65ddf9395b: Download complete
#fb30af17153: Download complete
#ffe54c5c139: Download complete
#1fbf306f8cb: Download complete
#b0d0f5795eeb: Download complete
#5d785144815: Download complete
#a900fcceb38: Download complete
#785b9873bd4: Download complete
#66cdc02555d: Download complete
#e911c59da28: Download complete
#ef9446ba5c: Download complete
#5e184b986a: Download complete
#219d58cc3f9: Download complete
#53aff88babcb4: Download complete
#79b28aef1a10: Download complete
#700e8e7d617: Download complete
#0e280e9aa8c: Download complete
#72f92e29533: Download complete
#b69c76bd2b6b: Download complete
#5aa2a3a6539: Download complete
#706ed7d4ce0ea: Download complete
#9a90e45e52c: Download complete
#4f4fb700ef54: Already exists
#6f8fab512d1: Download complete
#2fc4187e3b4: Download complete
Digest: sha256:2a7c2b20d118ba92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
Status: Downloaded newer image for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest
PS C:\Windows\system32>
```

```

Administrator: Windows PowerShell
PS C:\Windows\system32> docker run --name nagosdemo -p 8888:80 jasonrivers/nagios:latest
Adding password for user nagiosadmin
chown: warning: '.' should be '::': 'nagios.nagios'
Started runsvdir, PID is 13
checking permissions for nagios & nagiosgraph
rsyslogd: [origin software="rsyslogd" swVersion="8.2312.0" x-pid="33" x-info="https://www.rsyslog.com"] start

Nagios Core 4.5.7
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-10-24
License: GPL

Website: https://www.nagios.org
Nagios 4.5.7 starting... (PID=23)
Local time is Wed Dec 11 07:15:55 UTC 2024
wproc: Successfully registered manager as @wproc with query handler
nagios: Nagios 4.5.7 starting... (PID=23)
nagios: Local time is Wed Dec 11 07:15:55 UTC 2024
nagios: LOG VERSION: 2.0
nagios: qh: Socket '/opt/nagios/var/rw/magios.qh' successfully initialized
nagios: qh: core query handler registered
nagios: qh: echo service query handler registered
nagios: qh: help for the query handler registered
nagios: wproc: Successfully registered manager as @wproc with query handler
nagios: wproc: Registry request: name=Core Worker 44;pid=44
nagios: wproc: Registry request: name=Core Worker 44;pid=44
nagios: wproc: Registry request: name=Core Worker 42;pid=42:nagios: wproc: Registry request: name=Core Worker 42;pid=42

wproc: Registry request: name=Core Worker 46;pid=46
nagios: wproc: Registry request: name=Core Worker 46;pid=46
wproc: Registry request: name=Core Worker 45;pid=45
nagios: wproc: Registry request: name=Core Worker 45;pid=45
wproc: Registry request: name=Core Worker 55;pid=55
nagios: wproc: Registry request: name=Core Worker 55;pid=55
wproc: Registry request: name=Core Worker 47;pid=47
nagios: wproc: Registry request: name=Core Worker 47;pid=47
wproc: Registry request: name=Core Worker 59;pid=59
nagios: wproc: Registry request: name=Core Worker 48;pid=48
wproc: Registry request: name=Core Worker 57;pid=57
nagios: wproc: Registry request: name=Core Worker 54;pid=54
wproc: Registry request: name=Core Worker 56;pid=56
wproc: Registry request: name=Core Worker 52;pid=52
nagios: wproc: Registry request: name=Core Worker 50;pid=50
nagios: wproc: Registry request: name=Core Worker 48;pid=48
nagios: wproc: Registry request: name=Core Worker 57;pid=57
nagios: wproc: Registry request: name=Core Worker 54;pid=54
nagios: wproc: Registry request: name=Core Worker 56;pid=56

localhost:8888
localhost:8888

Sign in
http://localhost:8888
Username: nagiosadmin
Password: *****
Sign in Cancel

Nagios
Version 4.5.7 October 24, 2024
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.8.

Dashboard
System Status
Latest Alerts
Server Statistics
Status Grid
Administrative Tasks
Page Tour

https://www.nagios.org/launch?utm_campaign=csp&utm_source=nagioscore...

```

The screenshot shows the Docker Desktop interface. The left sidebar has options: Containers (selected), Images, Volumes, Builds, Docker Scout, and Extensions. The main area is titled "Containers" with a search bar and a button to "Only show running containers". It displays three running containers:

	Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
<input type="checkbox"/>	nagosdemo	841817c079d0	jasonrivers/nagios	8888:80	N/A	5 minutes	 
<input type="checkbox"/>	quizzical_germa	83d4c8d33678	luxxx000/si		N/A	4 days ago	 
<input type="checkbox"/>	minikube	bc81bd6703d8	k8s-minikul	0.22 Show all ports (5)	N/A	5 days ago	 

Below the table, it says "Showing 3 items".

The "Walkthroughs" section contains two cards:

- Multi-container applications (8 mins)
- \$ docker init Containerize your application (3 mins)

At the bottom, status indicators show "Engine running" and "Kubernetes running". The PowerShell window shows the following commands:

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

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

PS C:\Windows\system32> docker stop nagosdemo
nagosdemo
PS C:\Windows\system32> docker rm nagosdemo
nagosdemo
PS C:\Windows\system32> docker rmi nagosdemo
Error response from daemon: No such image: nagosdemo:latest
PS C:\Windows\system32> docker rmi jasonrivers/nagios:latest
Untagged: jasonrivers/nagios:latest
Deleted: sha256:2a7c2b20d118baf92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
PS C:\Windows\system32>
```

Experiment 8:

- AWS ACADEMY LEARNING ACCOUNT CREATION
 - PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE
 - MAVEN WEB PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE
-
- AWS ACADEMY LEARNING ACCOUNT CREATION

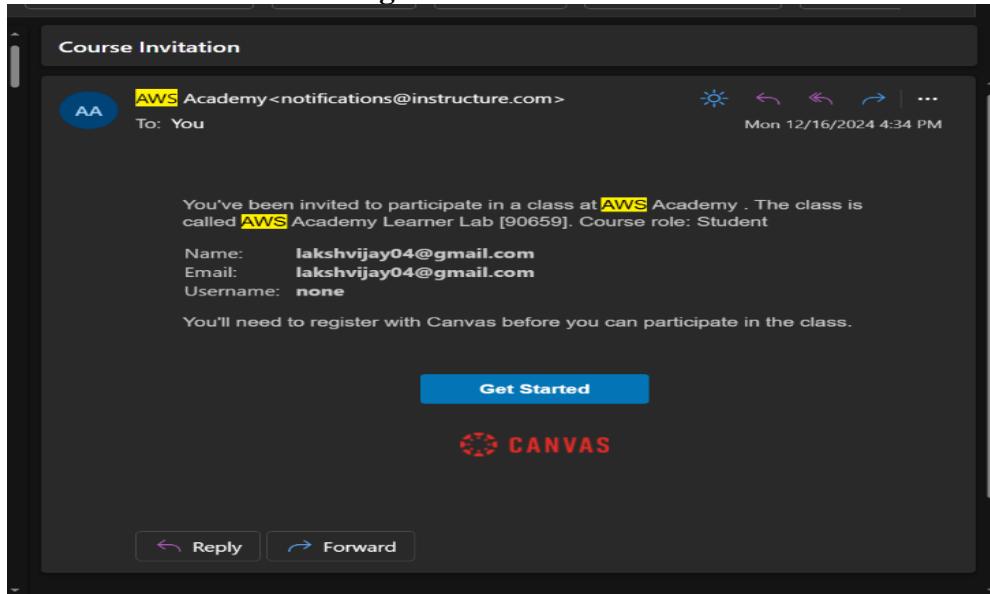
Go to your email and check for aws academy course invitation

Open it and click on create my account

Provide a password

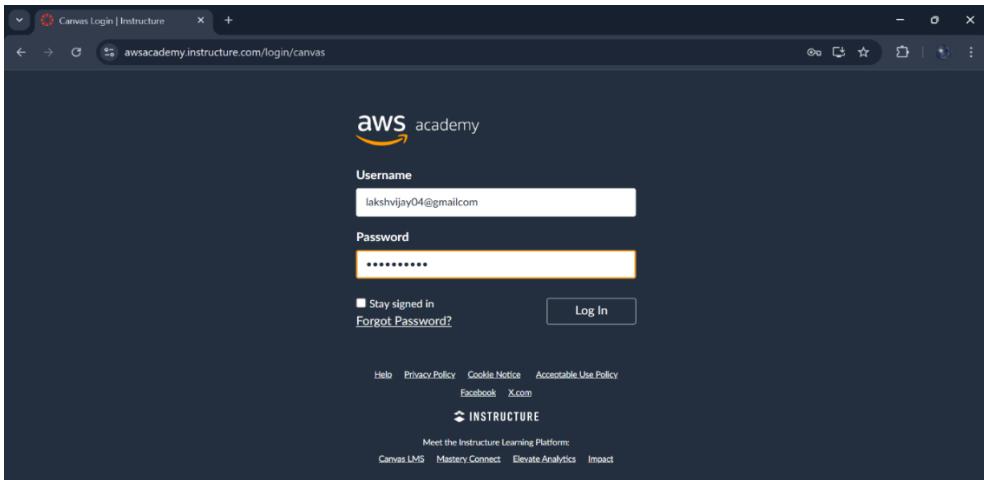
Select a time zone ,Check all the boxes

Scroll down and click on register.



Click on Student Login.

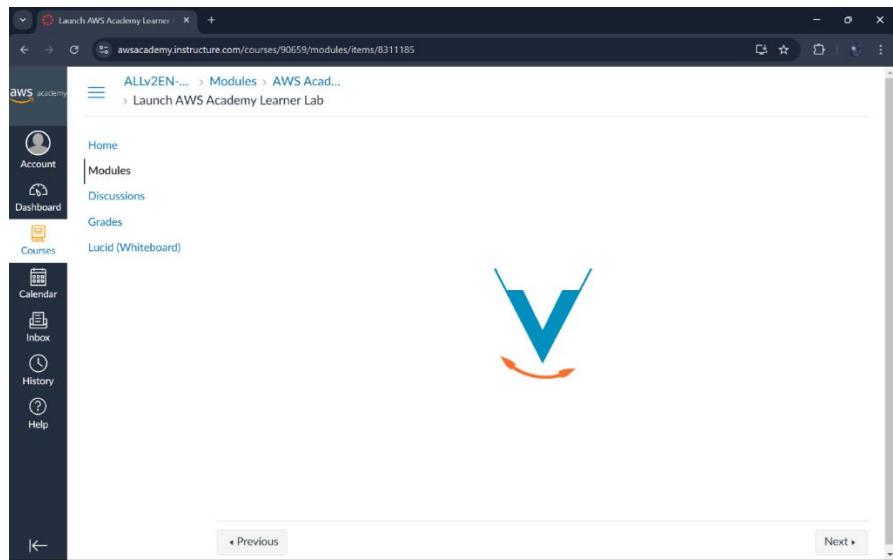
Fill the username and password and click on login



This is main page of AWS.

Click on the AWS academy learner lab:

Click on Modules.

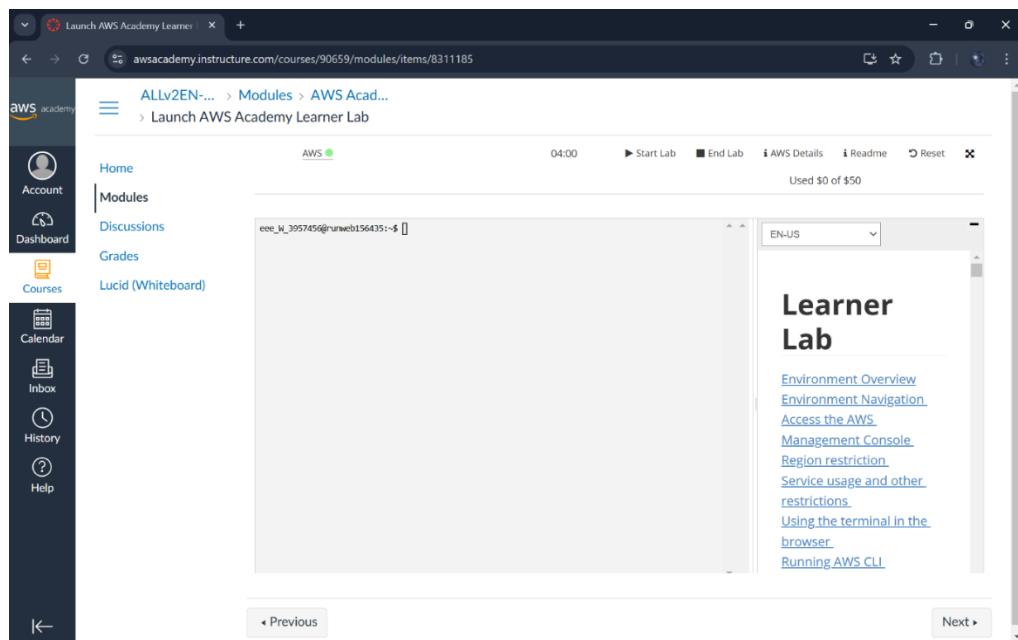


Scroll down and click on AWS academy learner lab.

Scroll down and agree to terms and conditions.

If there is a red dot beside AWS it means the sandbox is in stop state click on Start Lab.

Once you start the lab it will last for 4hours if you want to extend then comeback and start again.



You are given free 50\$, if you exceed you will not be able to access AWS with this email id again

The screenshot shows the AWS Console Home page for the US East (N. Virginia) region. On the left, there's a 'Service menu' box with a 'Next' button. Below it, a 'No recently visited services' section with a cube icon and a 'View all services' link. A 'Explore one of these commonly visited AWS services.' section lists EC2, S3, RDS, and Lambda. On the right, the 'Applications' section shows 0 applications with a 'Create application' button. At the bottom, there are 'Welcome to AWS', 'AWS Health', and 'Cost and usage' boxes, along with navigation links like 'CloudShell', 'Feedback', and 'Cookie preferences'.

Once your lab work is done ,click on End lab.

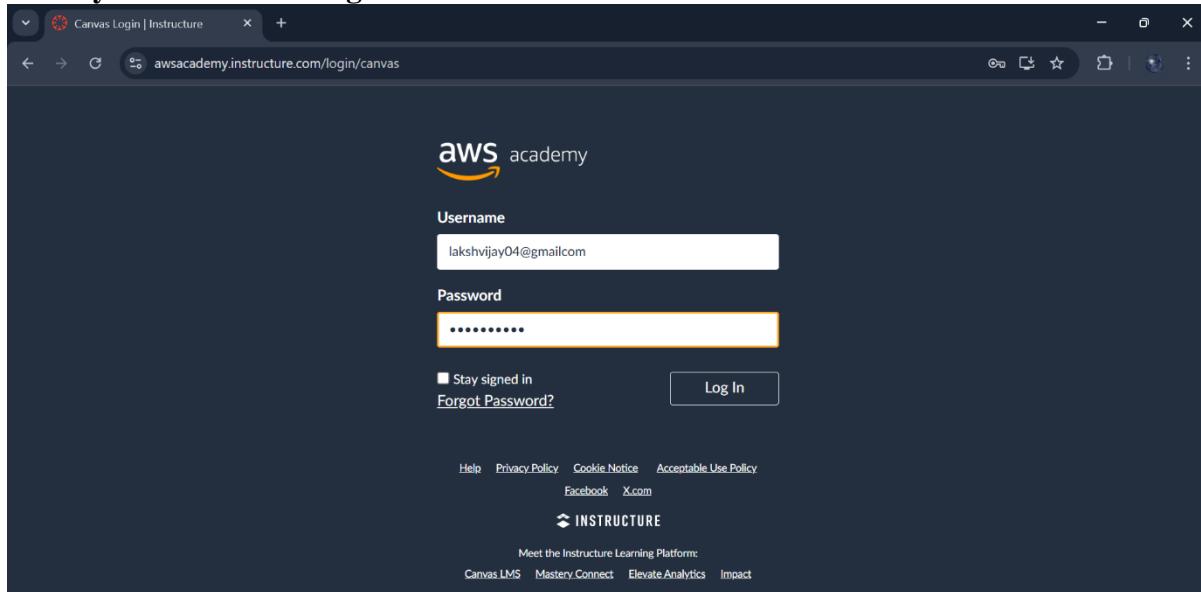
The screenshot shows the AWS Academy Learner Lab interface. The top navigation bar includes 'Launch AWS Academy Learner', 'Console Home', and 'awsacademy.instructure.com'. The main content area shows a breadcrumb path: 'ALLv2EN... > Modules > AWS Acad... > Launch AWS Academy Learner Lab'. A central modal dialog asks 'Are you sure you want to end the lab?' with 'Yes' and 'No' buttons. To the right, a sidebar titled 'Learner Lab' contains links to 'Environment Overview', 'Environment Navigation', 'Access the AWS Management Console', 'Region restriction', 'Service usage and other restrictions', 'Using the terminal in the browser', and 'Running AWS CLI'. Navigation arrows at the bottom allow switching between previous and next labs.

- o PROJECT DEPLOYMENT IN ASW CLOUD USING EC2 INSTANCE

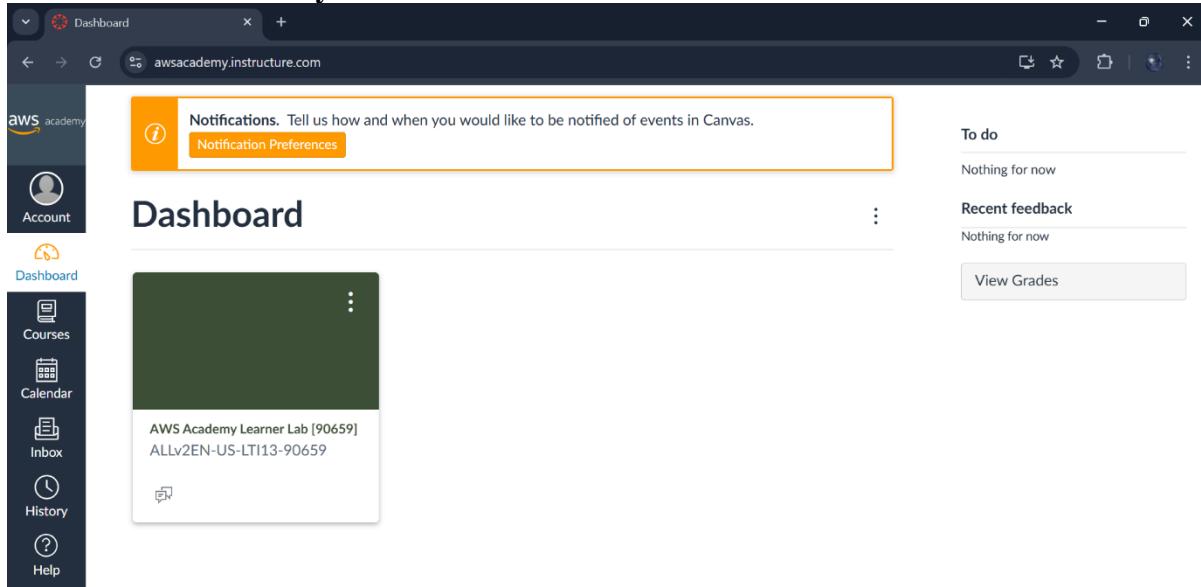
Search for aws Academy Login.

Click on Student Login.

Enter your details and login.



Click on AWS academy learner lab.



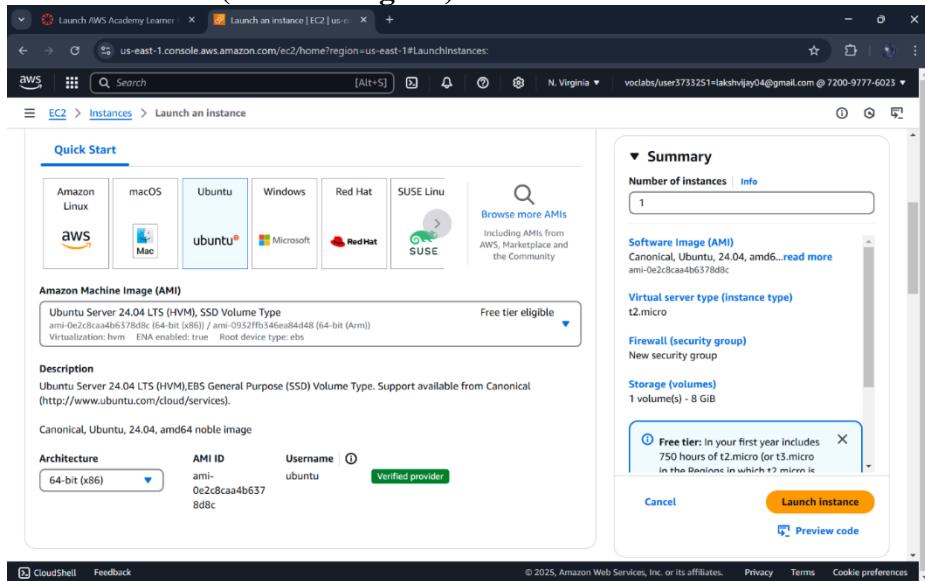
Click on Start lab.

After the dots turned green click on it.

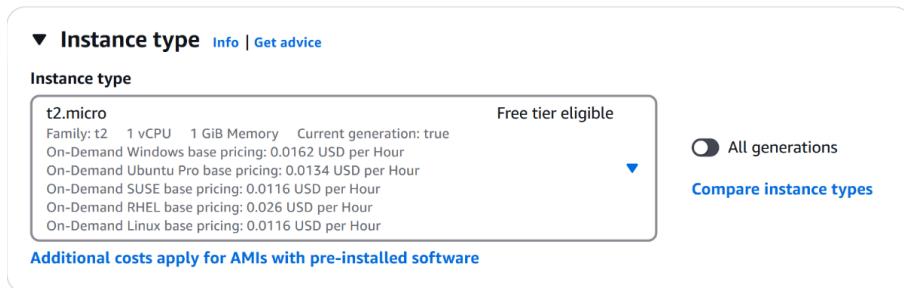
Click on EC2 and click on Launch Instance.

Enter a name like “MyWebAWS” to identify your server.

Choose Ubuntu(free tier eligible)

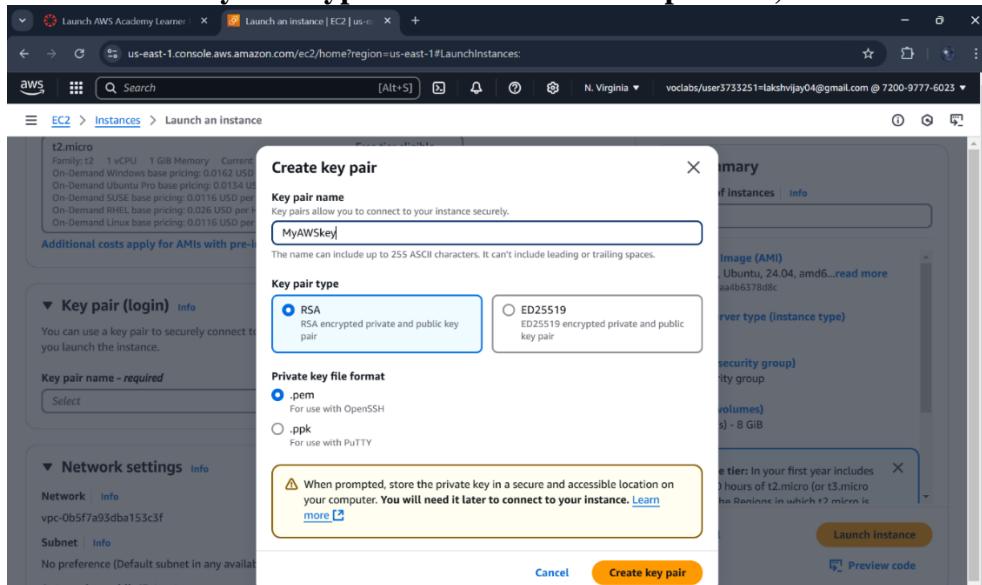


Instance type:Select t2.micro(1 CPU,1 GB RAM)



Create a new key pair.

Give a name to your keypair and download the .pem file, and save it securely.



Network: Enable Allow HTTP/HTTPS traffic to make your website accessible. Use the default 8 gb.

Network settings

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Regions in which t2.micro is available.

The instance is initiated successfully.

Success

Successfully initiated launch of instance (i-088d06be09cd58451)

Launch log

Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
MyAWSweb	i-088d06be09cd58451	Running	t2.micro	Initializing	View alarms +

Select an instance

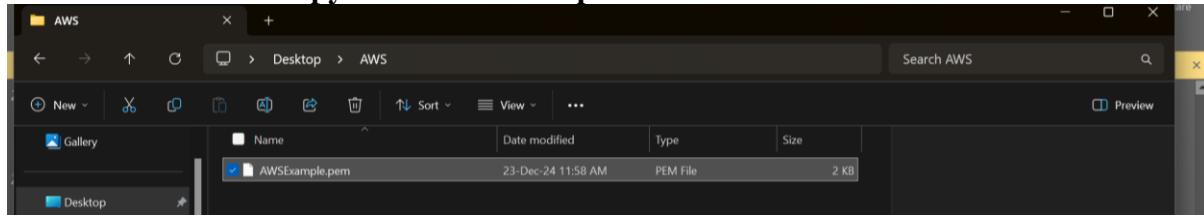
Select your instance, click Connect, and copy the SSH command under Example Heading.

The screenshot shows the AWS EC2 Instances page with the instance i-088d06be09cd58451 selected. The 'Connect to instance' section is open, and the 'SSH client' tab is active. It provides step-by-step instructions for connecting via SSH, including the command:

```
ssh -i "MyAWSkey.pem" ubuntu@ec2-52-90-233-190.compute-1.amazonaws.com
```

A note at the bottom states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username."

Create a folder and copy the downloaded.pem file in it.



Copy the path of the folder.

Open PowerShell on your computer and click on Run as Administrator.

Navigate to the folder where your .pem file is saved

Paste the ssh command and press enter

Type "yes" if prompted.

```
PS C:\Windows\system32> cd C:\Users\laksh\OneDrive\Desktop\SE-projects\AWS-key
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\AWS-key> ssh -i "MyAWSkey.pem" ubuntu@ec2-52-90-233-190.compute-1.amazonaws.com
The authenticity of host 'ec2-52-90-233-190.compute-1.amazonaws.com (52.90.233.190)' can't be established.
ED25519 key fingerprint is SHA256:fGqLxtNSXQ9a3f/FP6SrPHSXHLUwGeogIEGhYACbLNw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-90-233-190.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Thu Jan  2 05:09:45 UTC 2025

System load:  0.01      Processes:          106
Usage of /:   24.7% of 6.71GB  Users logged in:     0
Memory usage: 20%           IPv4 address for enx0: 172.31.25.214
Swap usage:   0%           IPv6 address for enx0: fe80::5031:25ff:fe25:214%enx0

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
```

Update the system to ensure using sudo apt update

```

ubuntu@ip-172-31-25-214 ~
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [764 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [173 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [965 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [238 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [310 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [108 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [111 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.0 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3844 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [949 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [562 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [208 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [112 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [108 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.7 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted/multiverse amd64 c-n-f Metadata [116 B]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [111 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7200 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [795 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [108 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [13.5 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [568 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [108 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [252 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/restricted/multiverse amd64 c-n-f Metadata [103 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2940 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 31,212 kB in 1min 10s (5058 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
58 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-25-214:~$
```

Install Docker to package and run on web application.

Sudo apt-get install docker.io

```

ubuntu@ip-172-31-25-214 ~
ubuntu@ip-172-31-25-214:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional package will be installed:
  bridge-utils contained dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools groupfs-mount | cgroup-lite debbootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEWER packages will be installed:
  bridge-utils contained dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 58 not upgraded.
Need to get 80.1 MB of archives.
After this operation, 384 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils 1.7.1~ubuntu [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 bridge-utils 1.7.1~ubuntu-0ubuntu4.2 [33.9 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data 2023112702-willysync1 [44590 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702-willysync1 [44590 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.98-2build2 [375 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 26.1.3~ubuntu-24.04.1 [32.4 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 80.1 MB in 1s (87.1 MB/s)
Preconfiguring packages...
Selecting previously unselected package pigz.
(Reading database ... 70601 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1~ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1~ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.12~ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12~ubuntu3.1) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../3-dnsmasq-base_2.90-2build2_amd64.deb ...
Unpacking dnsmasq-base (2.90-2build2) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_26.1.3~ubuntu1~24.04.1_amd64.deb ...
Unpacking docker.io (26.1.3~ubuntu1~24.04.1) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
```

Install git to manage and download code.

Sudo apt install git

Install nano for editing files directly on the server.

Sudo apt install nano

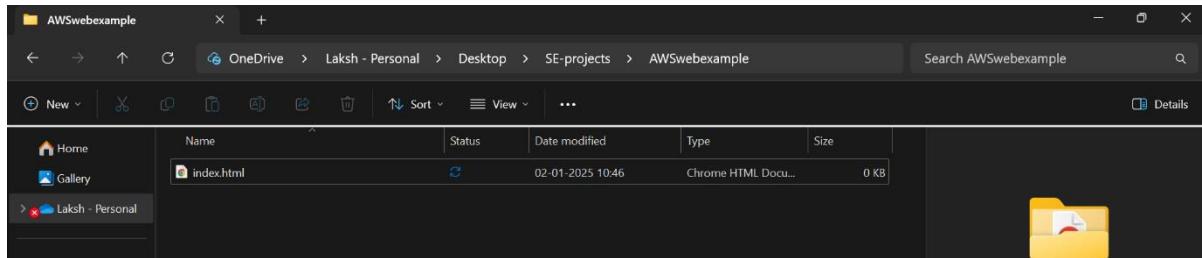
```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-22-36:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-22-36:~$ sudo apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-0ubuntu0.1).
nano set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-22-36:~$
```

Create your Web application

In this step ,we will build a simple page upload it to Github.

Create a folder in your Desktop.



On your computer,create a file named index.html and add the following content:
Hello from AWS

A screenshot of a code editor window titled 'index.html'. The code is as follows:

```
C: > Users > laksh > OneDrive > Desktop > SE-projects > AWSwebexample > index.html > html
1 <html>
2   <head><title>My Web Project</title></head>
3   <body>
4     <h1>Hello from AWS!</h1>
5   </body>
6 </html>
```

The code editor has a dark theme and includes icons for file operations like Open, Save, and Find.

Open your github account->Create new repository.

A screenshot of a GitHub repository page for 'lkhsh/AWSexample'. The repository is public and contains one branch ('main') and one commit ('First commit' by 'lkhsh'). The commit message is 'First commit' and it was made 4 minutes ago. On the right side of the page, there are sections for 'About' (with a note 'No description, website, or topics provided.'), 'Activity' (0 stars, 1 watching, 0 forks), 'Releases' (0 releases published, 'Create a new release'), and 'Packages' (0 packages published, 'Publish your first package'). At the bottom, there's a button to 'Add a README'.

Initialize git folder in it.

git add . to stage all changes

Git commit -m "First Commit" to commit the changes

```
MINGW64:/c/Users/laksh/OneDrive/Desktop/SE-projects/AWSwebexample
Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (master)
$ git init
Initialized empty Git repository in C:/Users/laksh/OneDrive/Desktop/SE-proj-
AWSwebexample/.git/
Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (master)
$ git add .

Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (master)
$ git commit -m "First commit"
[master (root-commit) efc857d] First commit
 1 file changed, 6 insertions(+)
 create mode 100644 index.html

Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (master)
$ git branch -M main

Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (main)
$ git remote add origin https://github.com/lxksh/AWSexample.git

Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lxksh/AWSexample.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Taksh@luxx MINGW64 ~/OneDrive/Desktop/SE-projects/AWSwebexample (main)
$
```

On the EC2 instance, clone your GitHub repository:

git clone "your repo URL"

Check if repo is cloned or not by typing ls.

Change your directory to the repo.

Build Docker container

Sudo docker build -t mywebexample

```
ubuntu@ip-172-31-25-214:~
1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Jan  2 05:09:47 2025 from 152.59.195.107
ubuntu@ip-172-31-25-214:~$ sudo git --version
git version 2.43.0
ubuntu@ip-172-31-25-214:~$ sudo docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1-24.04.1
ubuntu@ip-172-31-25-214:~$ git clone https://github.com/lxksh/AWSexample.git
Cloning into 'AWSexample'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-25-214:~$ cd AWSexample
ubuntu@ip-172-31-25-214:~/AWSexample$ sudo docker build -t mywebexample .
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/

ubuntu@ip-172-31-25-214:~/AWSexample$ nano Dockerfile
ubuntu@ip-172-31-25-214:~/AWSexample$ sudo docker build -t mywebexample .
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 63.49kB
Step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
```

Add the above content to the file

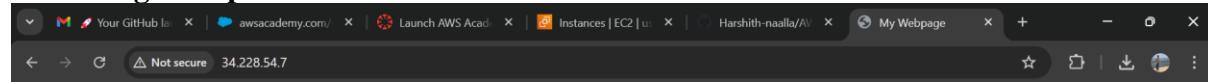
```
ubuntu@ip-172-31-29-33: ~/awsexample
GNU nano 7.2
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

Run the docker container to serve the web application:

Sudo docker run -d -p 80:80 mywebexample

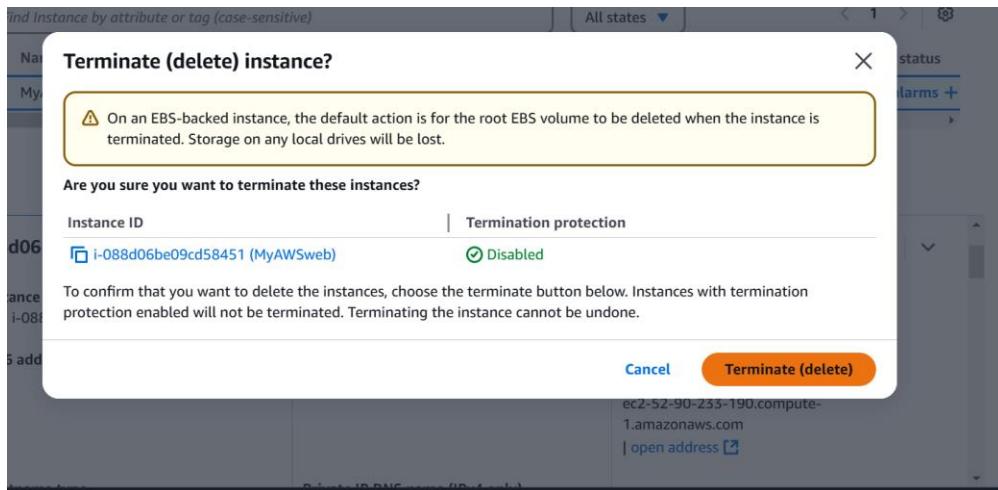
```
ubuntu@ip-172-31-25-214:~/AWSexample$ sudo run -d -p 80:80 mywebexample
sudo: run: command not found
ubuntu@ip-172-31-25-214:~/AWSexample$ sudo docker run -d -p 80:80 mywebexample
76ed61294320de380f6ce7461748dbfee01c1ebc5cb4e42cec029df89382150
ubuntu@ip-172-31-25-214:~/AWSexample$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
76ed61294320 mywebexample "/docker-entrypoint..." 15 seconds ago Up 14 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp eager_booth
ubuntu@ip-172-31-25-214:~/AWSexample$ sudo docker stop 76ed61294320
76ed61294320
```

You will get output like this.



Hello from AWS!

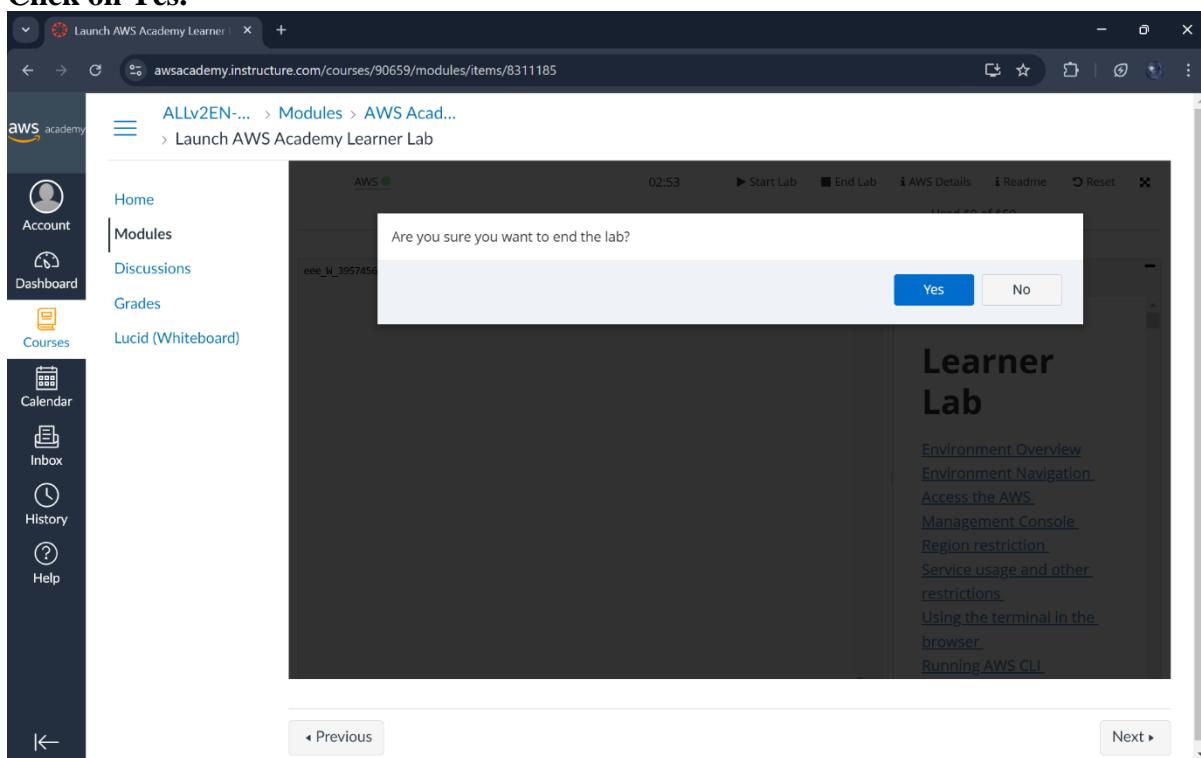
After running docker file stop it by using it's container ID
Terminate the EC2 instance in the AWS console by selecting it.



Check if the state is terminated or not under Instance State.

Click on End Lab

Click on Yes.

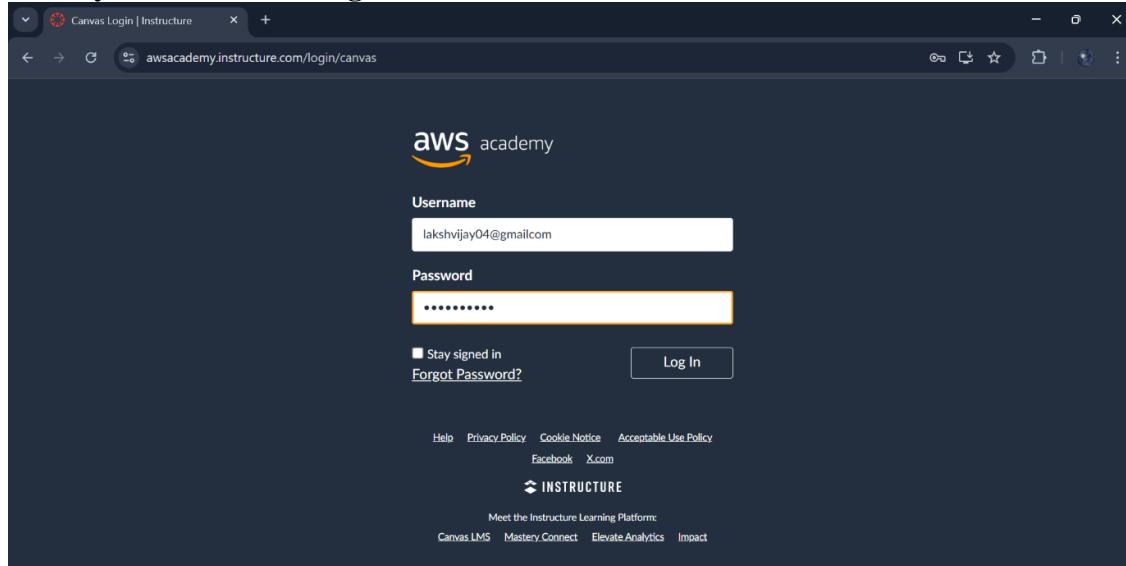


- MAVEN WEB PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE

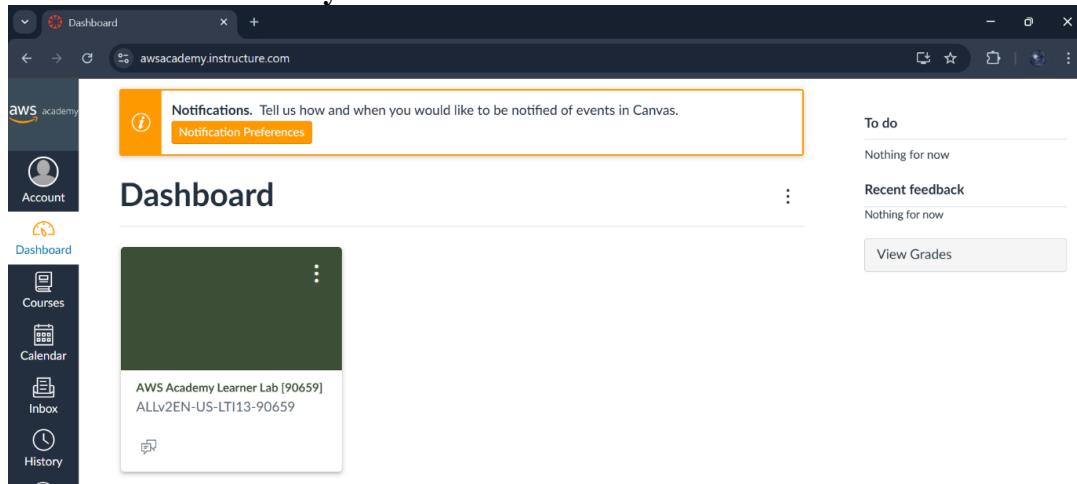
Search for AWS academy login

Click on Student Login.

Enter your details and login.

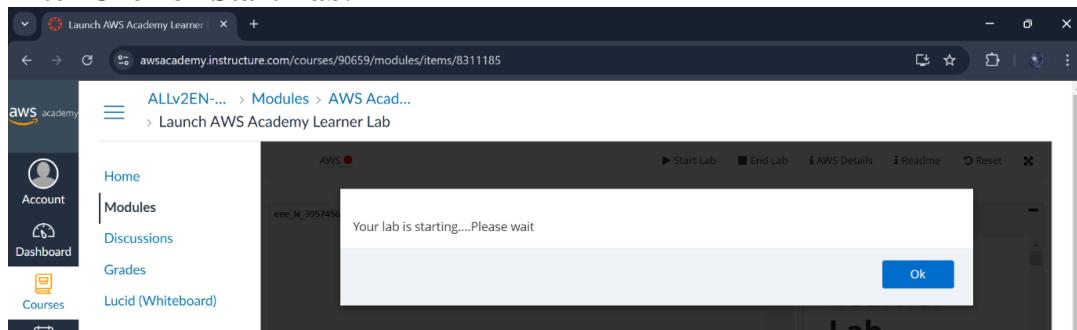


Click on AWS Academy Learner Lab ->Modules



Click on Launch AWS Academy Learner Lab.

After Click on Start Lab.



After it became green click on it.

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with navigation links like Home, Modules, Discussions, Grades, and Courses. The main area has a title 'ALLv2EN... > Modules > AWS Acad... > Launch AWS Academy Learner Lab'. Below this is a terminal window showing 'aws_395745@unweb15643 ~\$'. To the right is a 'Learner Lab' panel with a title 'Environment Overview' and several links: Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, and Running AWS CLI.

Click on EC2.
Click Launch Instance.

The screenshot shows the AWS EC2 service dashboard. On the left, there are navigation links for Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security. The main area features a summary table with metrics like Key pairs (1), Load balancers (0), Placement groups (0), Security groups (1), Snapshots (0), and Volumes (0). Below this is a 'Launch instance' button and a 'Migrate a server' button. To the right, there's a 'Service health' section showing the AWS Health Dashboard, Region (US East (N. Virginia)), and Status (operating normally). There's also an 'Additional information' section with links to Getting started guide, Documentation, All EC2 resources, Forums, Pricing, and Contact us.

Enter a descriptive name:ex:MavenWebProjectServer

The screenshot shows the 'Launch an instance' wizard. In the 'Summary' step, there's a 'Cancel' button and a prominent orange 'Launch instance' button. Below these are 'Preview code' and 'Add additional tags' buttons. On the left, there's a 'Name and tags' section where 'MavenWebServer' is entered. Below that is an 'Application and OS Images (Amazon Machine Image)' section with a search bar and a list of recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and SUSE. At the bottom, there are 'Recent' and 'Quick Start' tabs, and a 'CloudShell' and 'Feedback' button.

AMI: Select Ubuntu Server(Free Tire Eligible)

Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible
ami-0e2c8caa4b6378d8c (64-bit (x86)) / ami-0932fb1546ea8d48 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

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

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Username
64-bit (x86)	ami-0e2c8caa4b6378d8c	ubuntu

Verified provider

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Instance type:Choose t2.micro

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

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Key pair:Create a key pair or use an existing one.Save the .pem file securely

▼ Key pair (login) [Info](#)

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

Key pair name - required

MavenAWSkey

[Create new key pair](#)

Network settings:Enable Allow HTTP/HTTPS traffic

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

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:

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

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

Configure storage Info Advanced
1x 8 GiB gp3 Root volume 3000 IOPS (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Instance is initiated.

Success
Successfully initiated launch of instance (i-0c712ca5bae0c4a74)

Launch log

Wait for the status change to running

Instances (2) Info Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
MyAWSweb	i-088d06be09cd58451	Terminated	t2.micro	-	View alarms +
MavenWebSer...	i-0c712ca5bae0c4a74	Running	t2.micro	Initializing	View alarms +

Select your instance, click Connect ,and copy the SSH command under Example Heading.

Create a folder and copy the download .pem file in it.

Open on your computer->Run as administrator

Navigate to the folder where you have saved.pem file

Paste the SSH command and press enter.

If prompted,type “yes” to confirm the connection.

```
ubuntu@ip-172-31-26-193:~ 
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Windows\system32> cd C:\Users\laksh\OneDrive\Desktop\SE-projects\AWSMaven-key
PS C:\Users\laksh\OneDrive\Desktop\SE-projects\AWSMaven-key> ssh -i "MavenAWSkey.pem" ubuntu@ec2-54-225-15-174.compute-1.amazonaws.com
The authenticity of host 'ec2-54-225-15-174.compute-1.amazonaws.com (54.225.15.174)' can't be established.
ED25519 key fingerprint is SHA256:WVtg4HMuqJUleYYbi/byOkDGWxRr6h24j0BXfg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-225-15-174.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Thu Jan  2 06:22:23 UTC 2025

System load:  0.11      Processes:          106
Usage of /:   24.7% of 6.71GB  Users logged in:     0
Memory usage: 21%           IPv4 address for enx0: 172.31.26.193
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

Update the system:sudo apt update

```
ubuntu@ip-172-31-26-193:~$ sudo apt update
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[...]
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [126 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [572 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [387.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [301.0 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [764 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [707 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [965 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [238 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [310 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [574 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [503 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.0 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3844 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [288 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main Translation-en [28 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [16.7 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.7 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Components [212 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 kB]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 kB]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7209 B]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [111 kB]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [795 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [169 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.6 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [15.5 kB]
[...]
[38 {8 Components amd64 store 0 B} | 43 Commands amd64 store 1397 B / 13.5 kB 10%]
```

Install Docker:sudo apt-get install docker.io -y

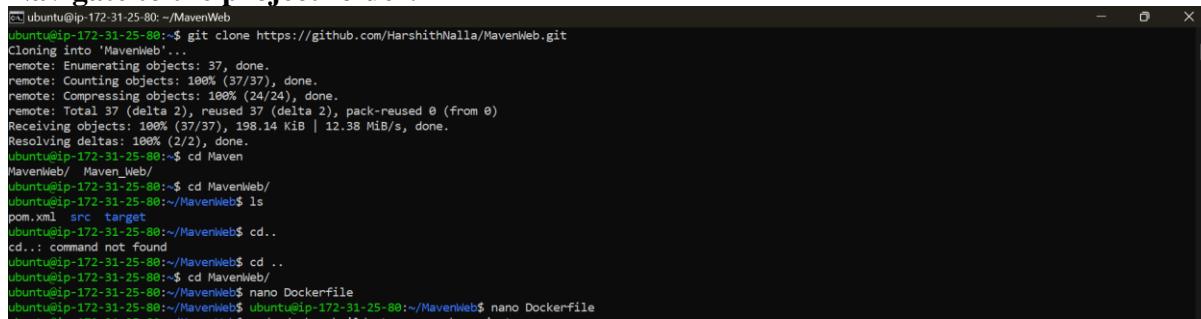
```
ubuntu@ip-172-31-26-193:~$ 
Reading state information... Done
58 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-26-193:~$ sudo apt-get install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils contained dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils contained dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 58 not upgraded.
Need to get 80.1 MB of archives.
After this operation, 304 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/units/main amd64 containerd amd64 1.7.19+really1.7.12-0ubuntu4.2 [38.6 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702-willsync1 [4450 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 26.1.3-0ubuntu1~24.04.1 [32.4 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 80.1 MB in 47.6 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 70601 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.19+really1.7.12-0ubuntu4.2_amd64.deb ...
Unpacking containerd (1.7.19+really1.7.12-0ubuntu4.2) ...
[...]
```

Install git:sudo apt install git -y

Install Nano:sudo apt install nano -y

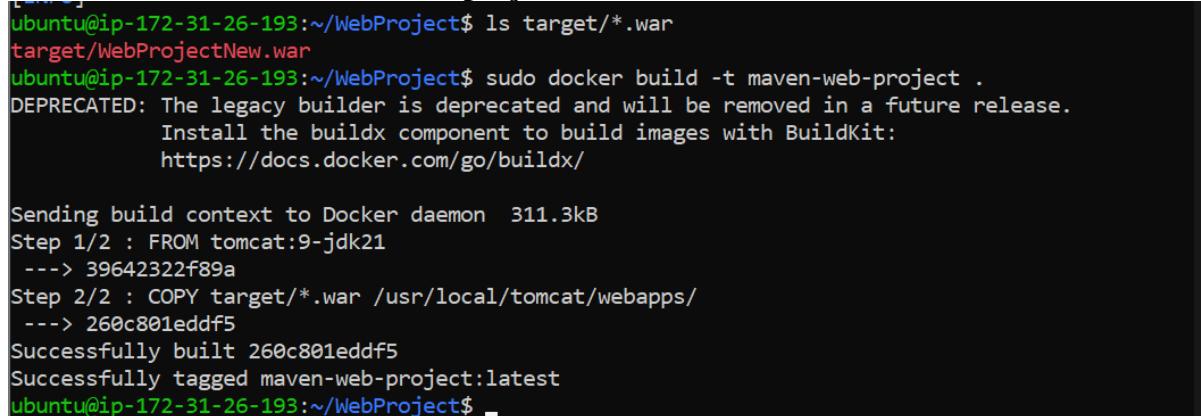
```
ubuntu@ip-172-31-26-193:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-26-193:~$ sudo apt install nano -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-2ubuntu0.1).
nano set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-26-193:~$
```

Go to your github repository and copy your URL.
Clone your repository
Navigate to the project folder.



```
ubuntu@ip-172-31-25-80:~/MavenWeb$ git clone https://github.com/HarshithNalla/MavenWeb.git
Cloning into 'MavenWeb'...
remote: Enumerating objects: 37, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 37 (delta 2), reused 37 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (37/37), 198.14 KiB | 12.38 MiB/s, done.
Resolving deltas: 100% (2/2), done.
ubuntu@ip-172-31-25-80:~/MavenWeb$ cd MavenWeb
MavenWeb$ Maven_Web/
ubuntu@ip-172-31-25-80:~/MavenWeb$ ls
pom.xml  src  target
ubuntu@ip-172-31-25-80:~/MavenWeb$ cd..
cd..: command not found
ubuntu@ip-172-31-25-80:~/MavenWeb$ cd ..
ubuntu@ip-172-31-25-80:~/MavenWeb$ ls
Dockerfile
ubuntu@ip-172-31-25-80:~/MavenWeb$ ubuntu@ip-172-31-25-80:~/MavenWeb$ nano Dockerfile
```

Build docker image
Sudo docker build -t maven-web-project



```
ubuntu@ip-172-31-26-193:~/WebProject$ ls target/*.war
target/WebProjectNew.war
ubuntu@ip-172-31-26-193:~/WebProject$ sudo docker build -t maven-web-project .
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 311.3kB
Step 1/2 : FROM tomcat:9-jdk21
--> 39642322f89a
Step 2/2 : COPY target/*.war /usr/local/tomcat/webapps/
--> 260c801eddf5
Successfully built 260c801eddf5
Successfully tagged maven-web-project:latest
ubuntu@ip-172-31-26-193:~/WebProject$
```



```
GNU nano 7.2                               Dockerfile
FROM tomcat:9-jdk21
COPY target/*.war /usr/local/tomcat/webapps/
```

Add the following content based on JDK version used during development.

Run your docker image.

```
successfully tagged maven-web-project:latest
ubuntu@ip-172-31-26-193:~/WebProject$ sudo docker run -d -p 9090:8080 maven-web-project
9d7ac564b80baccb2fc04b21a8f1d746664a753dc414f16defde975da82bed69
ubuntu@ip-172-31-26-193:~/WebProject$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
9d7ac564b80b        maven-web-project   "catalina.sh run"   14 seconds ago    Up 12 seconds     0.0.0.0:9090->8080/tcp, :::9090->8080/tcp   boring_bhabha
```

In the AWS EC2 dashboard, go to security and click the Security group ID

The screenshot shows the AWS EC2 Security Groups page. A green notification box at the top says: "Inbound security group rules successfully modified on security group (sg-0479a81bae8c4d36f | launch-wizard-2)". Below it is a "Details" link. The main table lists three security groups:

Name	Security group ID	Security group name
-	sg-07b34de87ec8914ac	launch-wizard-1
-	sg-0479a81bae8c4d36f	launch-wizard-2
-	sg-0681ceb38d21a47eb	default

Add an inbound rule.

The screenshot shows the "Edit inbound rules" page for the security group "sg-0479a81bae8c4d36f - launch-wizard-2". It displays two inbound rules:

- Inbound rule 1:**
 - Security group rule ID:** sgr-0e69b8da5445d6105
 - Type:** Custom TCP
 - Port range:** 9090
 - Source:** 0.0.0.0/0
- Inbound rule 2:**
 - Type:** Custom TCP
 - Port range:** 9090
 - Source:** 0.0.0.0/0

The changes are successfully saved.

Inbound rules (4)

Manage tags

Edit inbound rules

Search

Name	Security group rule ID	IP version
-	sgr-0e69b8da5445d6105	IPv4
-	sgr-0961e869bd7aecb0c	IPv4
-	sgr-0e1f4c580aba47fd6	IPv4
-	sgr-0b35d5dab7e93d496	IPv4

CloudShell Feedback Privacy Terms Cookie preferences © 2025, Amazon Web Services, Inc. or its affiliates.

Copy the public IP address.

Public IPv4 address
52.207.229.40 | open address

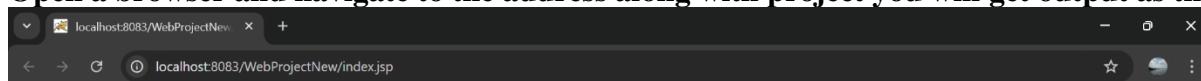
Private IPv4 addresses
172.31.26.193

Instance state
Shutting-down

Public IPv4 DNS
ec2-52-207-229-40.compute-

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Open a browser and navigate to the address along with project you will get output as this



Sudo docket ps to display all containers

Copy the container ID .Stop the docker container ID

In the EC2 dashboard go to instance state and select terminate instance

Launch AWS Academy Le Instances | FC2 | us-east- HTTP Status 404 – Not Found | WebProject/target/m2e- ChatGPT

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

aws Search [Alt+S]

EC2 > Security Groups sg-0479a81bae8c4d36f - launch-wizard-2

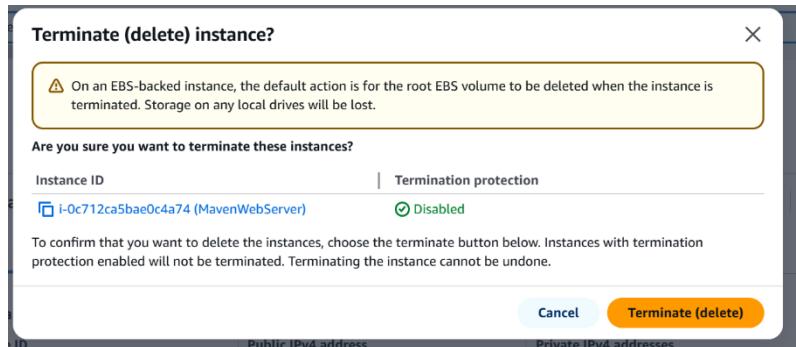
Instances (1/1) Info Last updated less than a minute ago Connect Instance state Actions Launch instances

Name	Instance ID	Instance state
favenWebSer...	i-0c712ca5bae0c4a74	Running

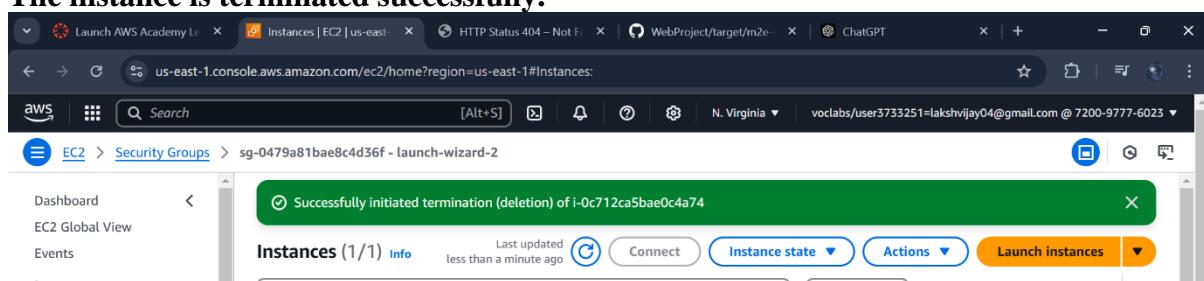
Stop instance Start instance Reboot instance Hibernate instance

Terminate (delete) instance

Click on Terminate



The instance is terminated successfully.



End lab to avoid charges.

Click on Yes

