

1 INSTALLATION OF STARUML, GIT BASH AND GITHUB ACCOUNT CREATION

GIT site to download Git Bash <https://git-scm.com>

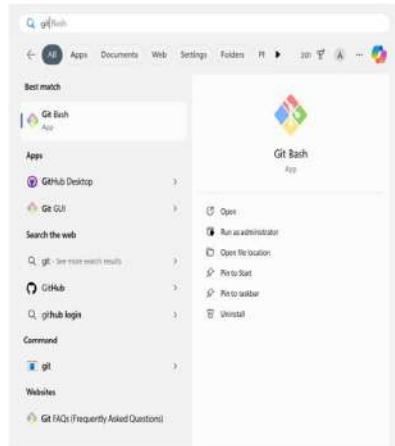


- Git Bash is an

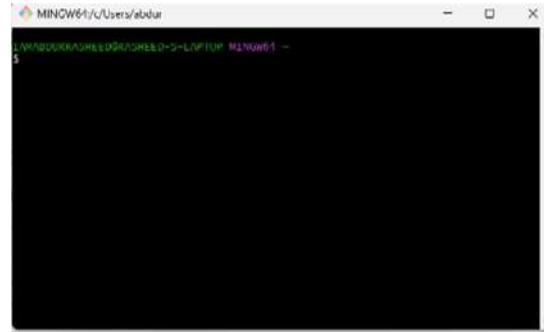
application that lets you use Git commands on your computer.

- **Git:** A tool for tracking changes in files, mainly used for managing code in software projects.
- **Bash:** A command-line tool that allows you to type in commands to perform various tasks on your computer.
- **Purpose:**
- Git Bash combines Git and Bash, so you can easily manage your code and use commands to work with files and projects. It's like a smart helper for keeping track of changes and collaborating with others on coding projects.

INSTALLED GitBash



GitBash INTERFACE AFTER INSTALLATION

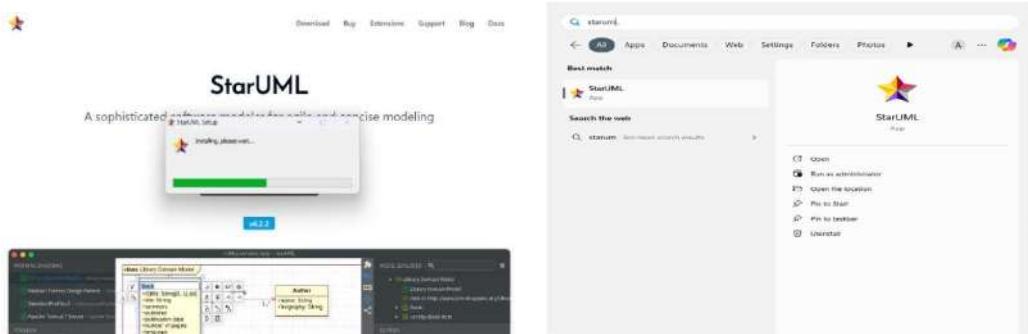


StarUML website to download StarUML

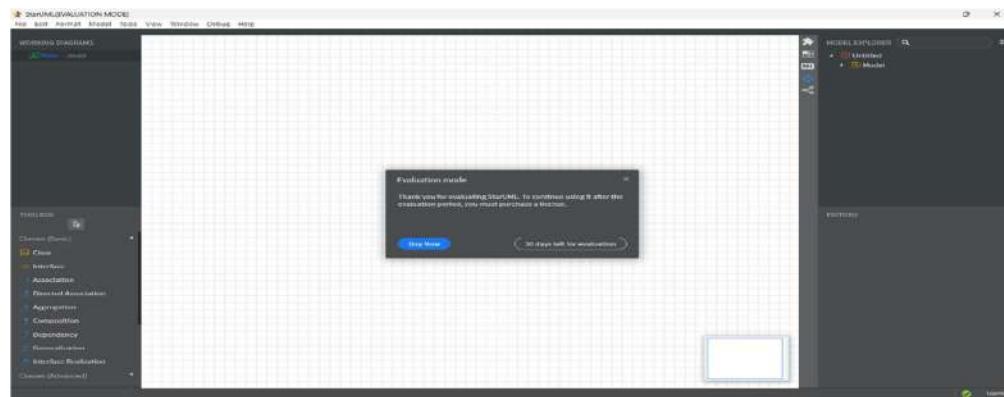


- StarUML is a software tool used to create diagrams that help in designing and understanding complex systems, especially in software development.
- It allows you to visually map out how different parts of a system or application will work together, making it easier to plan, communicate, and build the system effectively.
- Think of it as a drawing tool for software engineers to sketch out their ideas.

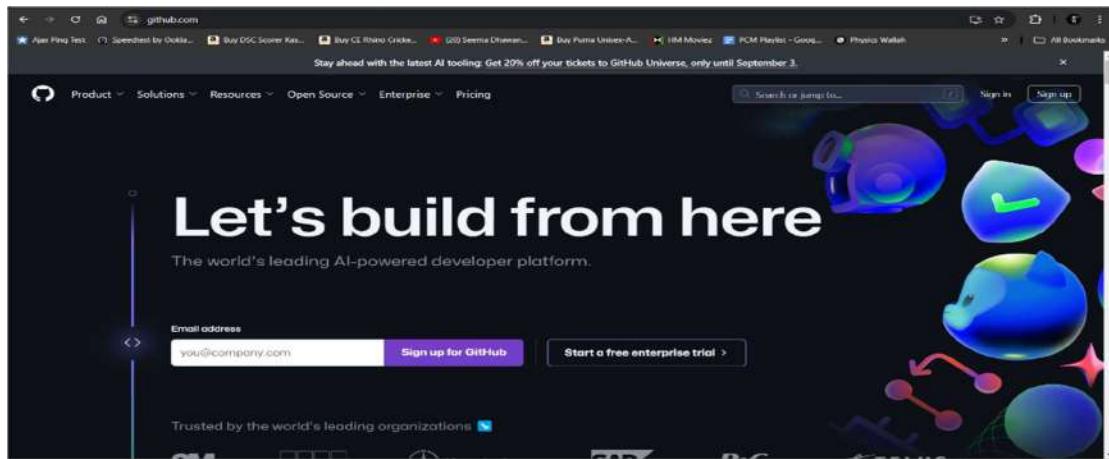
Installing StarUML



StarUML interface after installation



GitHub Website to sign in into it and create a GitHub Account



<https://github.com>

GitHub is a website where people can store and share their computer code.

- It's like a big, online folder where you can keep your work, make changes, and collaborate with others.
- Think of it as a place where you and your team can work on a project together, track changes, and keep everything organized.

My GitHub Dashboard After Creating an Account

The screenshot shows the GitHub dashboard. On the left, there is a sidebar with "Top repositories" (including "iamabdurrashid/iamabdurrashid", "RohitCodes/twix", and "diyaranjan1259/gym-data-management"), a "New" button, and a "Find a repository..." search bar. The main area has a "Home" section with a "Start writing code" button and a "Create a new repository" button. It also features sections for "Repositories that need your help" (with entries for "touchnet/jdeb", "gradle/gradle", and "joacmatosilva/DateTimeExtensions"), "Latest changes" (with items for "Yesterday" and "Last week"), and "Get AI-based coding suggestions". A large "UNIVERSE'24" banner is visible on the right side of the dashboard.

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

Keshav Memorial Institute of Technology

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_F_Soft_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 stands 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)

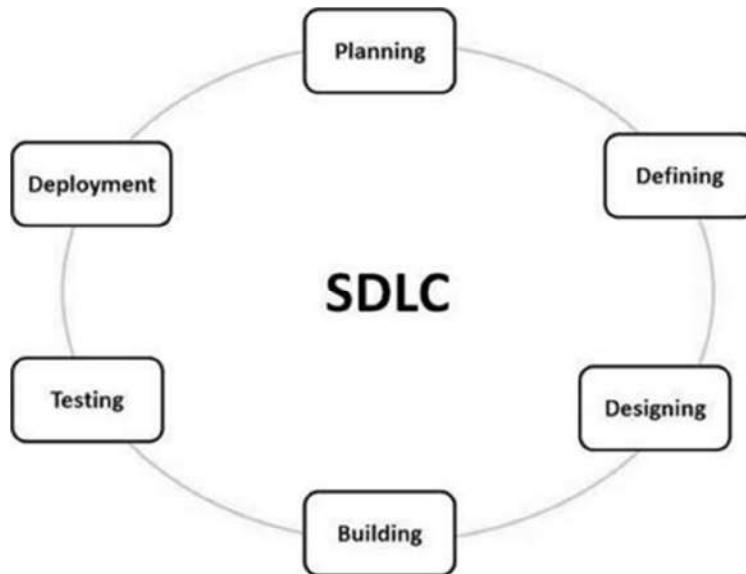
Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- 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.

SRS Document

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 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.
- 4. 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

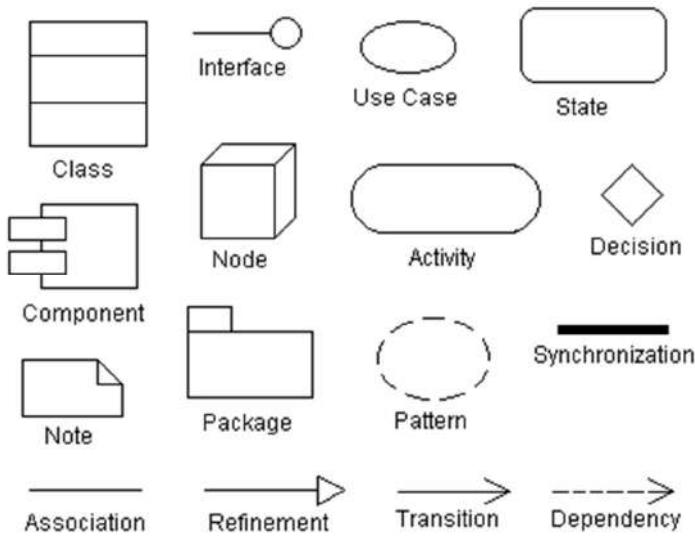
Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

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.

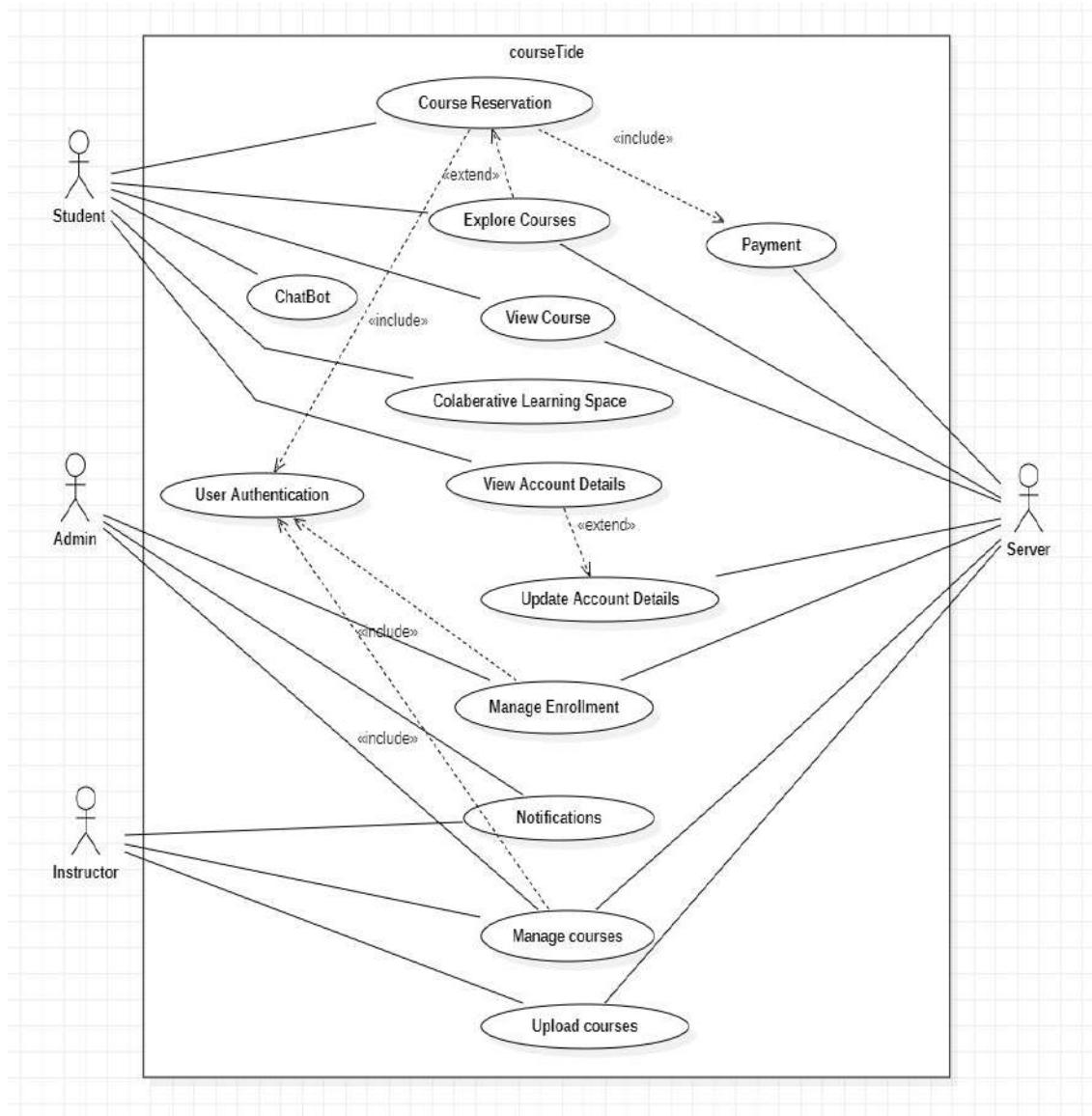
Appendix A: Glossary

- 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.

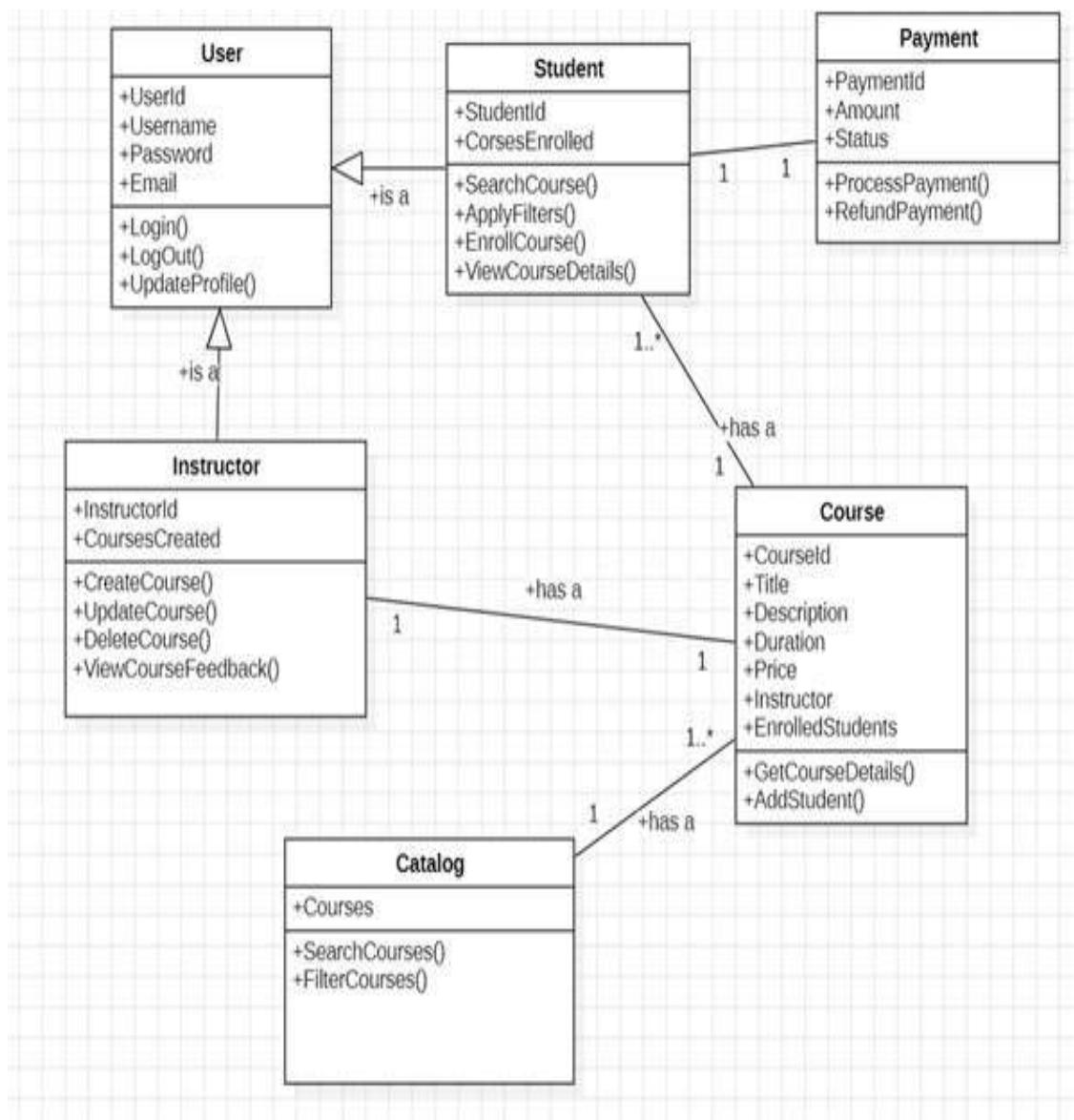
Appendix B: Analysis Models

- 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.

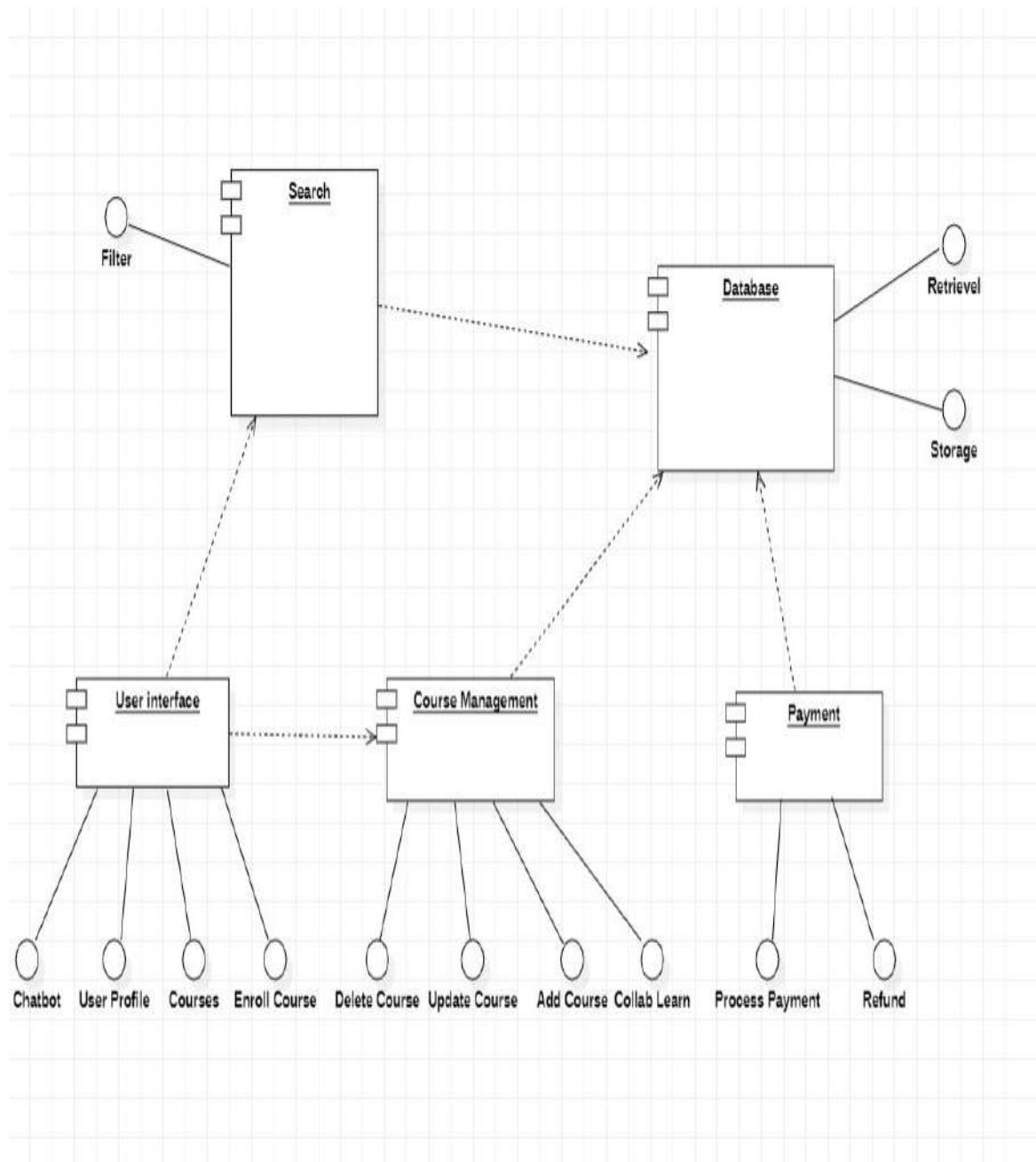
Use case diagram:



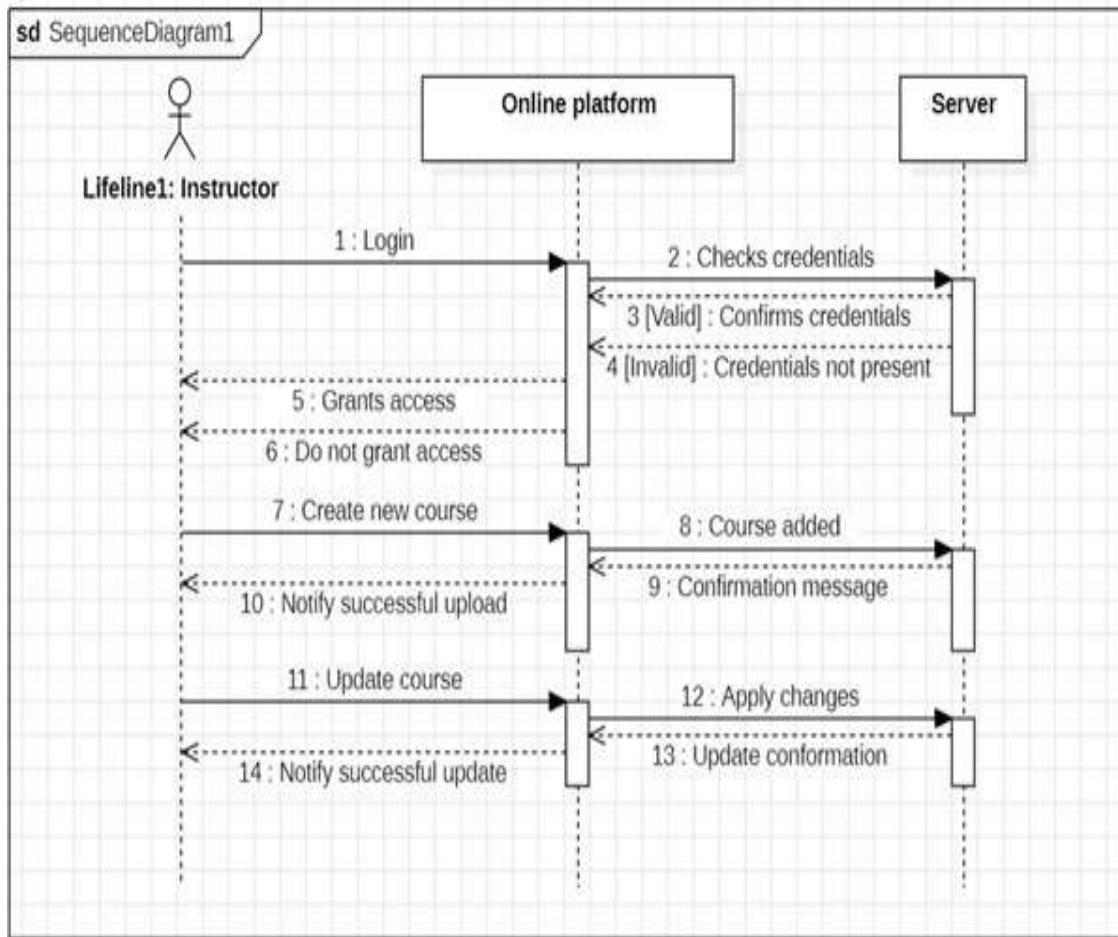
Class diagram:



Component diagram:



Sequence diagram:



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

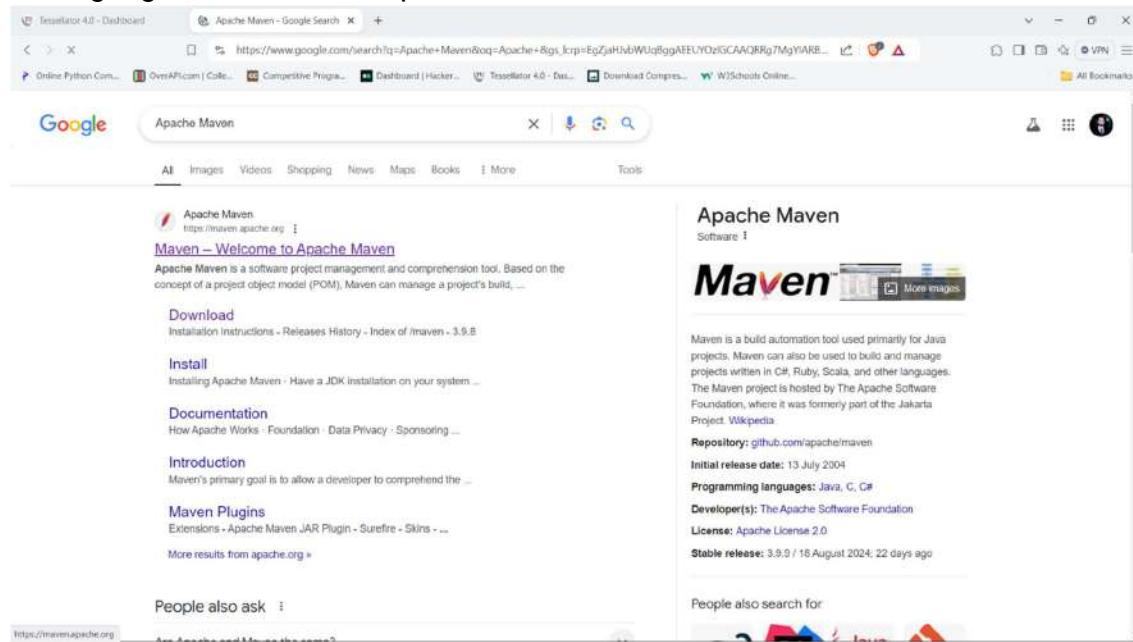
End Objective:	Discovering, analysing and recommending courses based on user preferences		
Created by:	1. Mohammed Abdur Rasheed 2. Nehith Sayini 3. Laksh Vijayvargiya 4. Jyothi Swaroop	On (date):	18-10-2024
User/Actor:	Administrator, student, instructor, server		
Trigger:	The user (admin, student, instructor) attempts to access the system		
Basic/Normal Flows			
User Actions	System Actions		
The user logs into the online course platform by entering valid credentials The user views their dashboard (Admin views course management, Instructor views enrolled students, Students view their enrolled courses). The user selects the View Course Content option The user edits their personal details or course information (if Admin or Instructor). The user enrolls in a course or creates a course (based on role).	The system displays a login page prompting the user to provide the correct username and password. The system checks the user's role (Admin, Instructor, or Student) and displays relevant features (e.g., course management, student lists, or enrolled courses). The system retrieves and displays the course content associated with the user's enrolled courses. The system provides editable fields for personal or course information and updates the database with the new details after confirming the changes. The system presents enrollment or course creation options and confirms the action once submitted.		

The user checks out and makes payments for course fees.	The system processes the payment (using online methods) and updates the billing records, providing a receipt for the course enrollment or creation fees.
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."

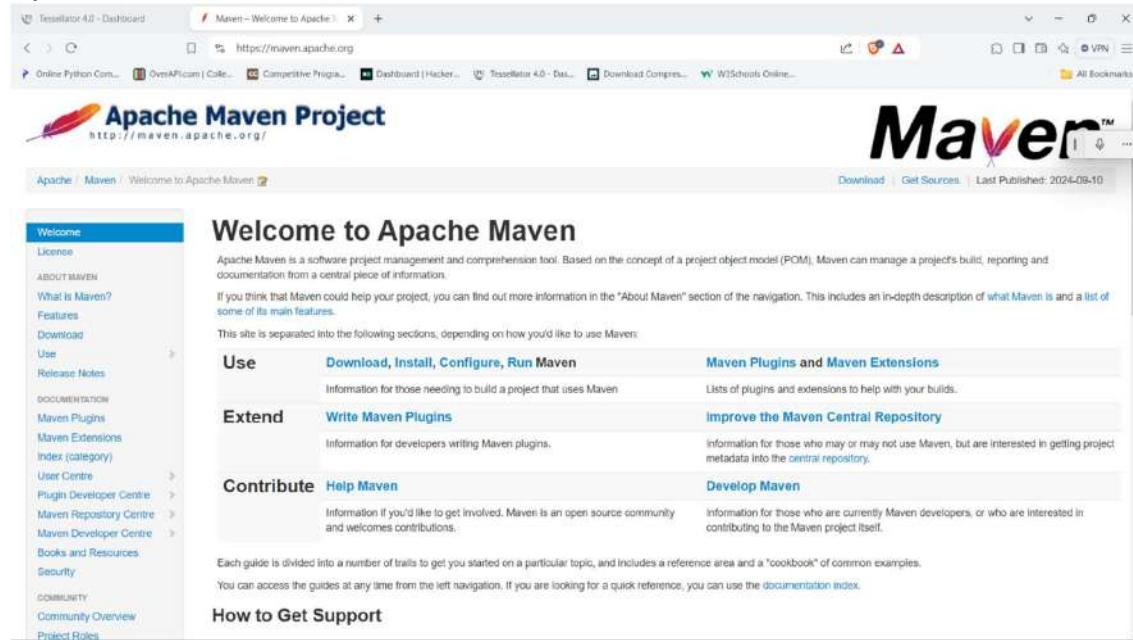
3. INSTALLATION OF ECLIPSE, MAVEN, JDK, TOMCAT, CONFIGURING TOMCAT TO ECLIPSE

INSTALLATION OF MAVEN

1. Go to google and search for Apache maven .



2. Open the first official link.



3. Click on apache-maven-3.9.9-bin.zip for downloading the zip file.

Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.9-bin.tar.gz.sha512	apache-maven-3.9.9-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.9-bin.zip	apache-maven-3.9.9-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.9-src.tar.gz.sha512	apache-maven-3.9.9-src.tar.gz.asc
Source zip archive	apache-maven-3.9.9-src.zip	apache-maven-3.9.9-src.zip.asc

Release Notes and Reference Documentation

- 3.9.9 Release Notes and Release Reference Documentation
- latest source code from source repository
- Distributed under the Apache License, version 2.0
- other:
 - all current release sources (plugins, shared libraries,...) available at <https://downloads.apache.org/maven/>

Other Releases

It is strongly recommended to use the latest release version of Apache Maven to take advantage of newest features and bug fixes. If you still want to use an old version, you can find more information in the [Maven Releases History](#) and can download files from the [Maven 3 archives](#) for versions 3.0.4+ and [legacy archives](#) for earlier releases.

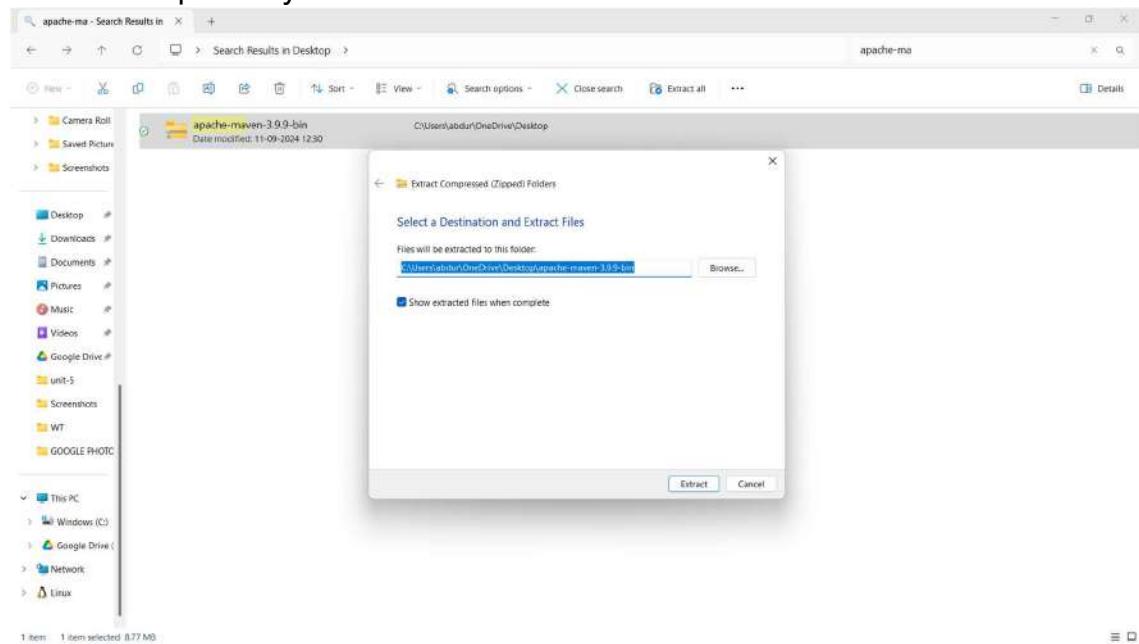
Previous Stable 3.8.x Release

Apache Maven 3.8.8 is the previous stable minor release for all users.

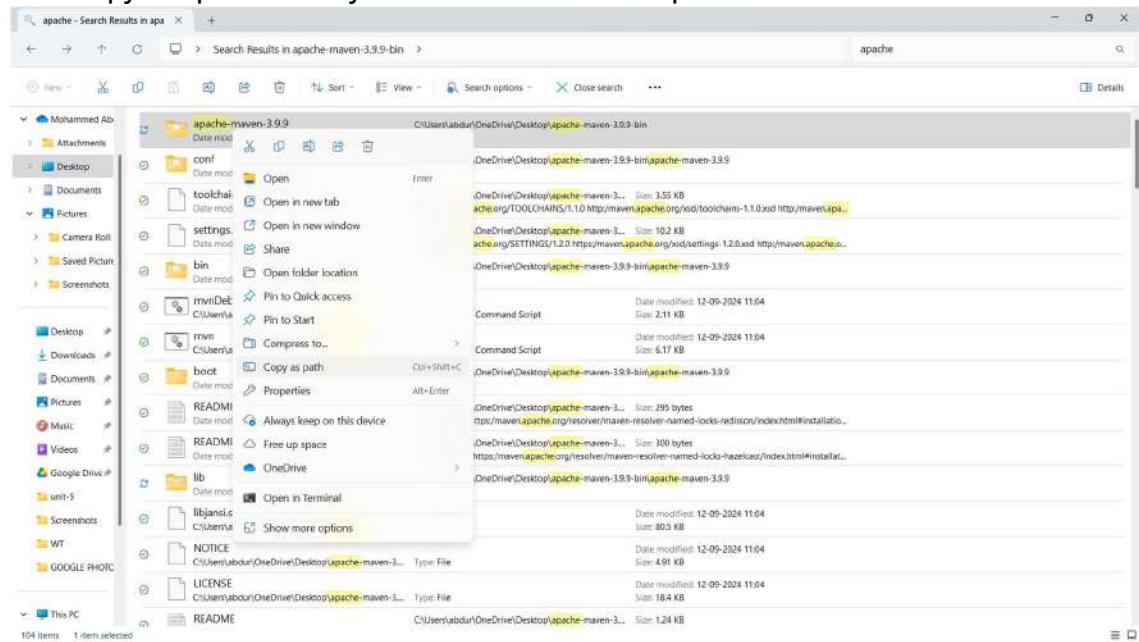
Java Development Kit (JDK) Maven 3.8+ requires JDK 1.7 or above to execute. It still allows you to build against 1.3 and other JDK versions by using toolchains.

4. Open the downloads folder and search for downloaded zip file.

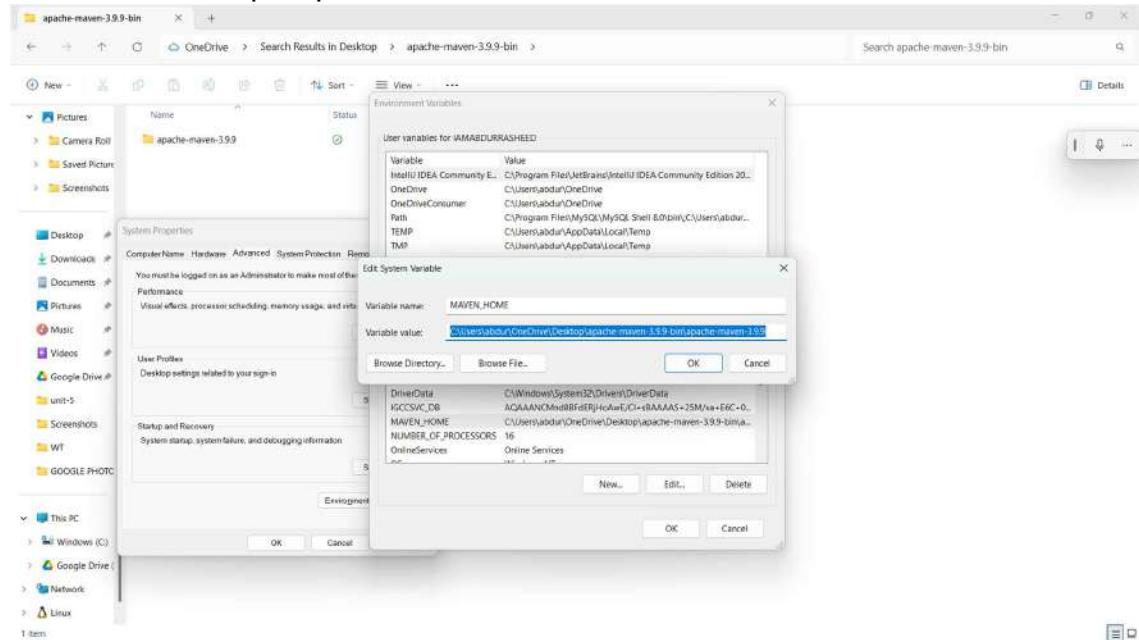
5. Extract the zip file in your desired folder.



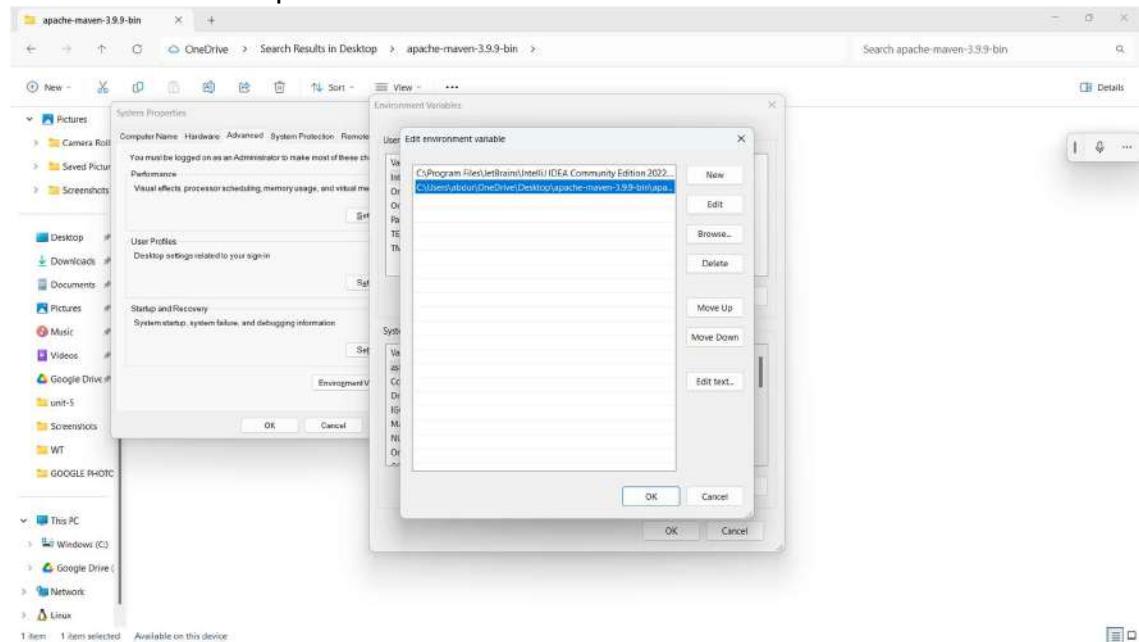
6. Now copy the path where you have extracted the zip file.



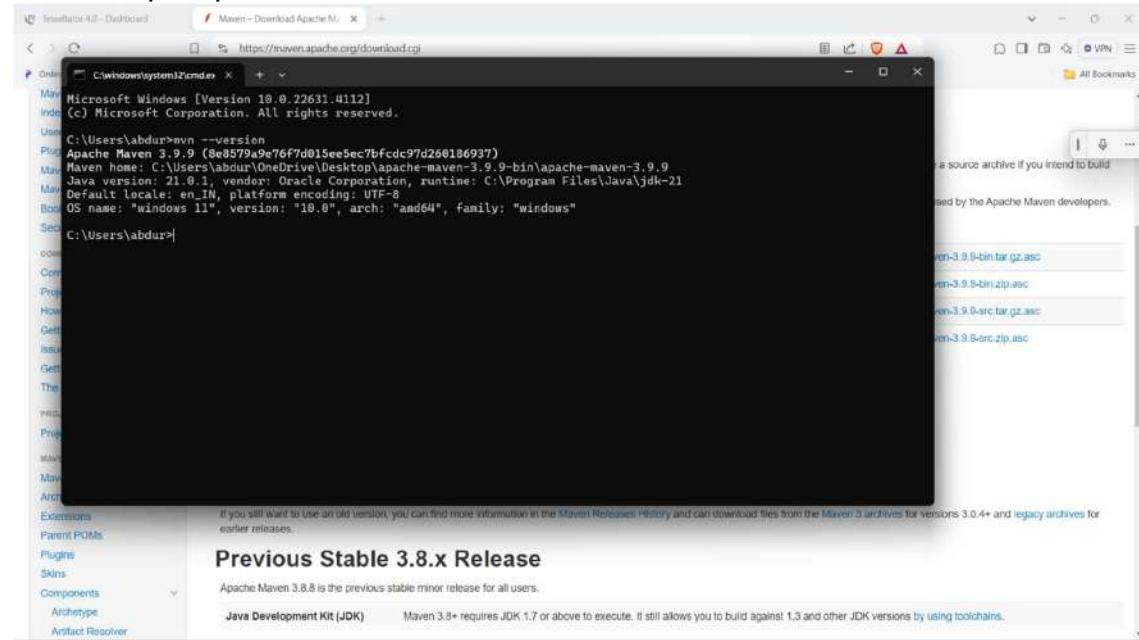
7. In System variables create a new system variable with name MAVEN_HOME and set value as the copied path.



8. Also add the maven path to the Environment Path variable.

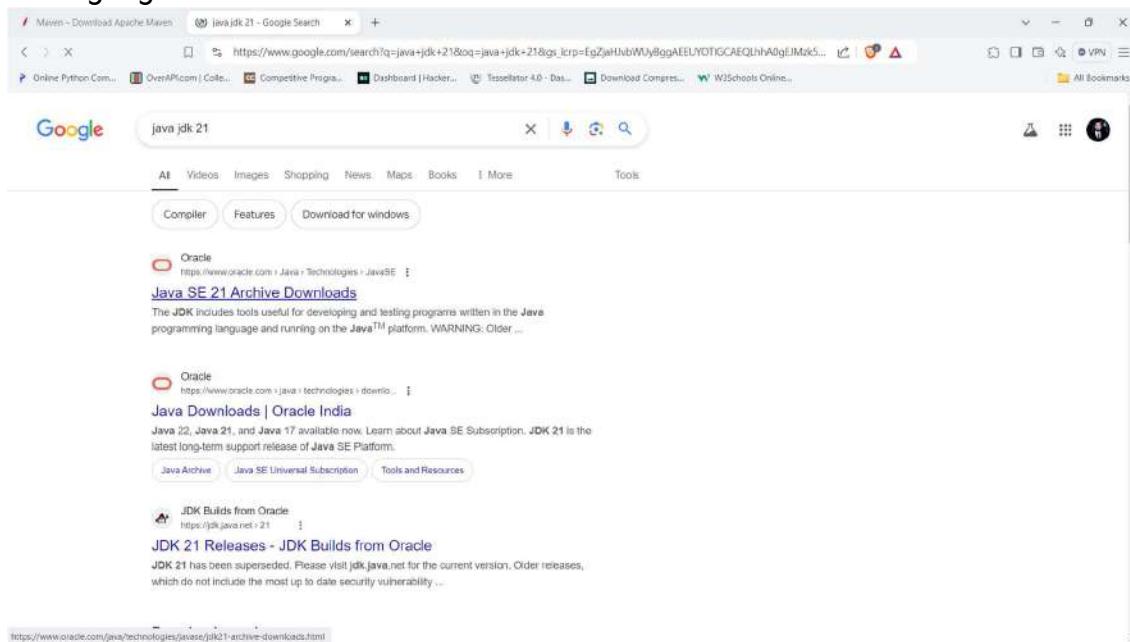


9. Check if maven is installed correctly by running “mvn -v” command in your command prompt.

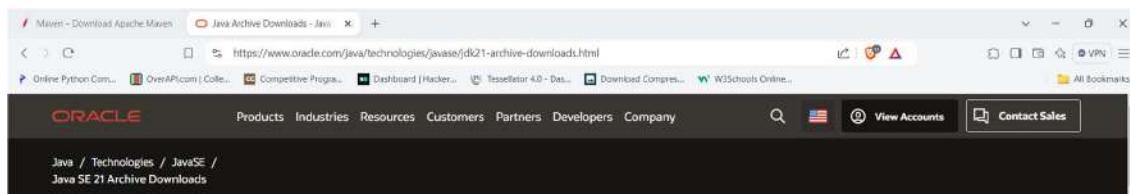


INSTALLATION OF jdk-21

1. Go to google and search for Java Jdk 21.



2. Visit the first official link of Oracle.



Java SE 21 Archive Downloads

Go to the Oracle Java Archive page.

The JDK is a development environment for building applications using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java™ platform.

WARNING: Older versions of the JDK are provided to help developers debug issues in older systems. They are not updated with the latest security patches and are not recommended for use in production.

For production use Oracle recommends downloading the latest JDK version.

Only developers and enterprise administrators should download these releases.

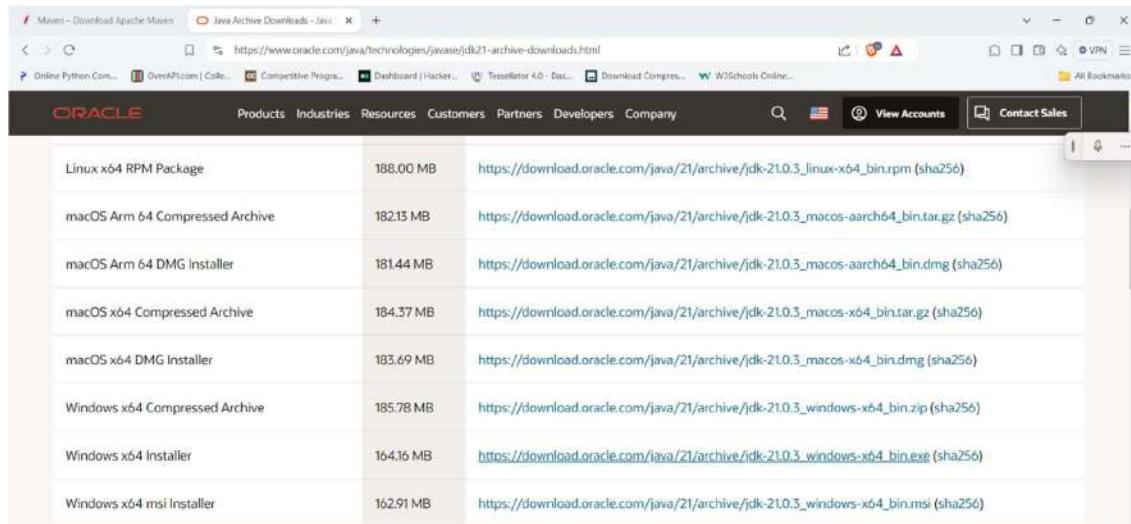
For current Java releases, please visit Oracle Java SE Downloads.

Java SE Development Kit 21.0.3

This software is licensed under the Oracle No-Fee Terms and Conditions License.

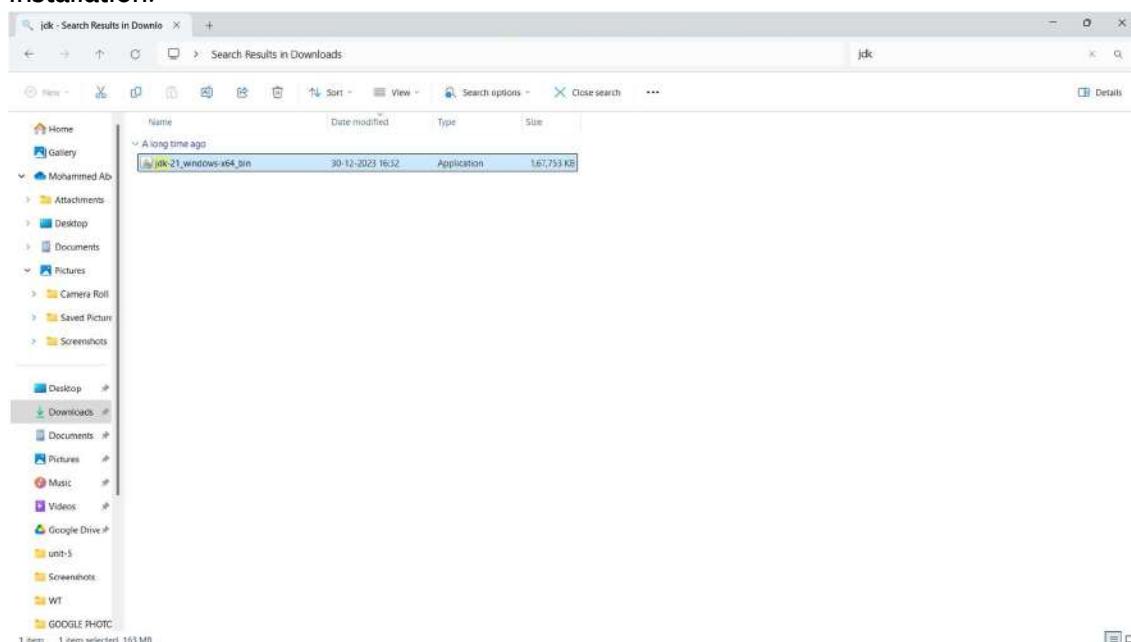
<https://www.oracle.com/countries-list.html#countries>

3. Download the appropriate version for your system. In my case it is Windows x64.

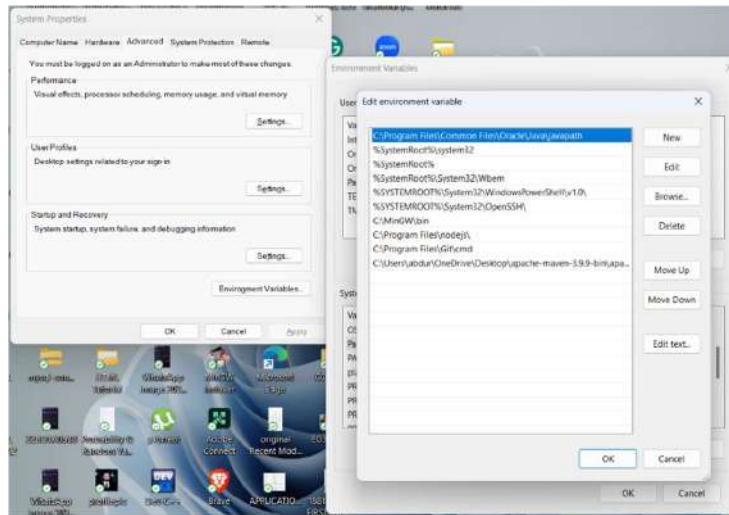


Linux x64 RPM Package	168.00 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_linux-x64_bin.rpm (sha256)
macOS Arm 64 Compressed Archive	182.13 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_macos-aarch64_bin.tar.gz (sha256)
macOS Arm 64 DMG Installer	181.44 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_macos-aarch64_bin.dmg (sha256)
macOS x64 Compressed Archive	184.37 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_macos-x64_bin.tar.gz (sha256)
macOS x64 DMG Installer	183.69 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_macos-x64_bin.dmg (sha256)
Windows x64 Compressed Archive	185.78 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_windows-x64_bin.zip (sha256)
Windows x64 Installer	164.16 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_windows-x64_bin.exe (sha256)
Windows x64 msi Installer	162.91 MB	https://download.oracle.com/java/21/archive/jdk-21.0.3_windows-x64_bin.msi (sha256)

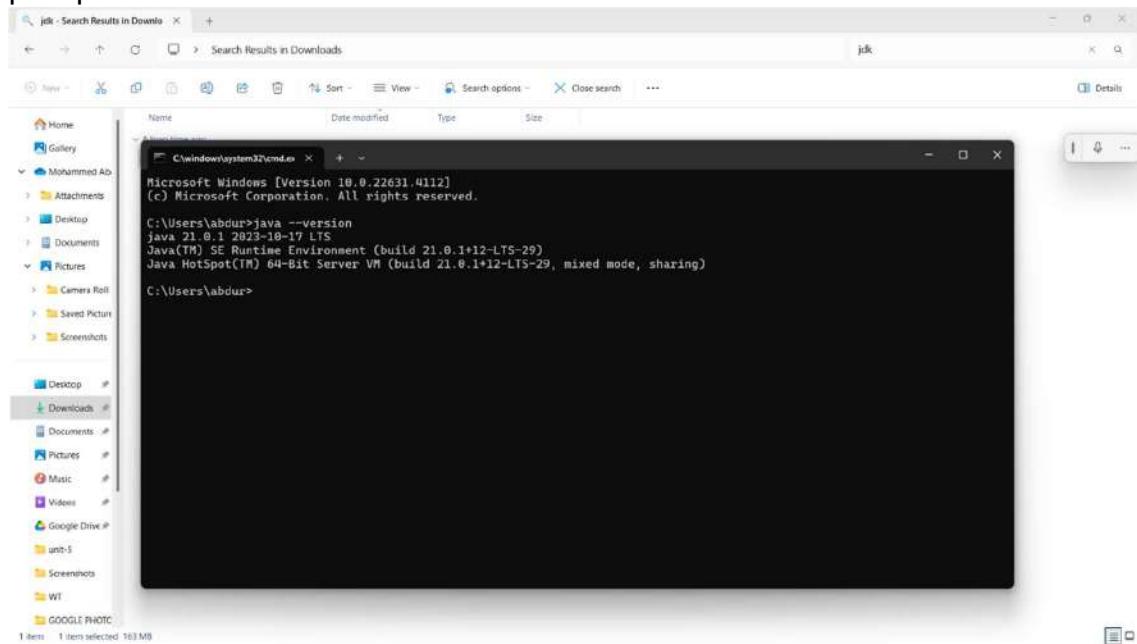
4. Open the downloads folder and then run the downloaded java file for installation.



5. Java Path in Environment Variables

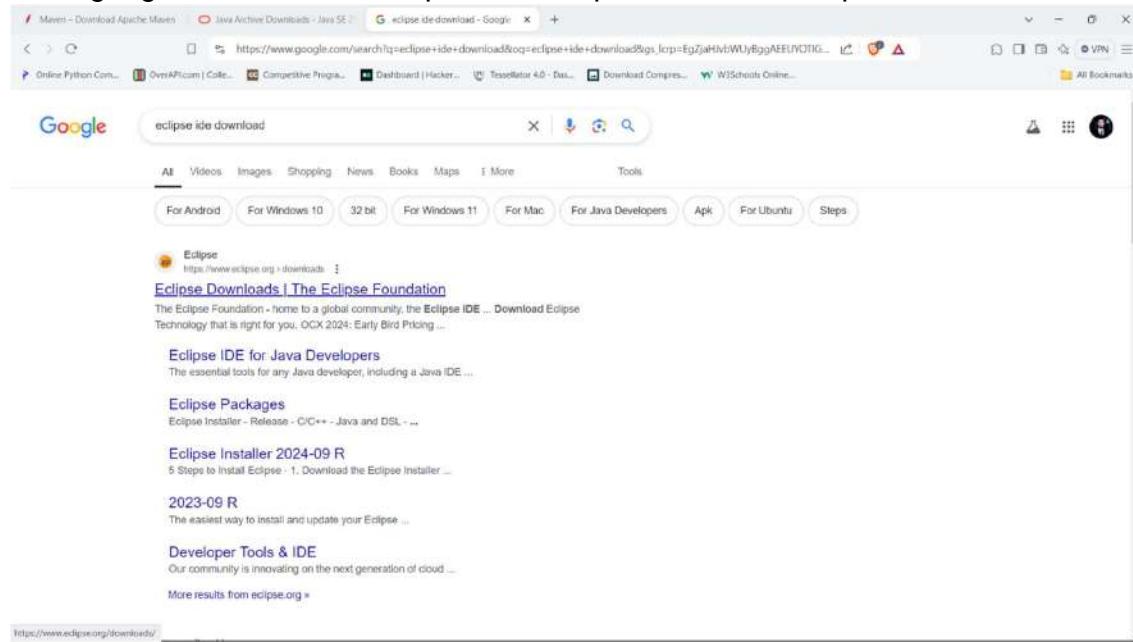


6. Verify the installation by running “java –version” command in your command prompt.

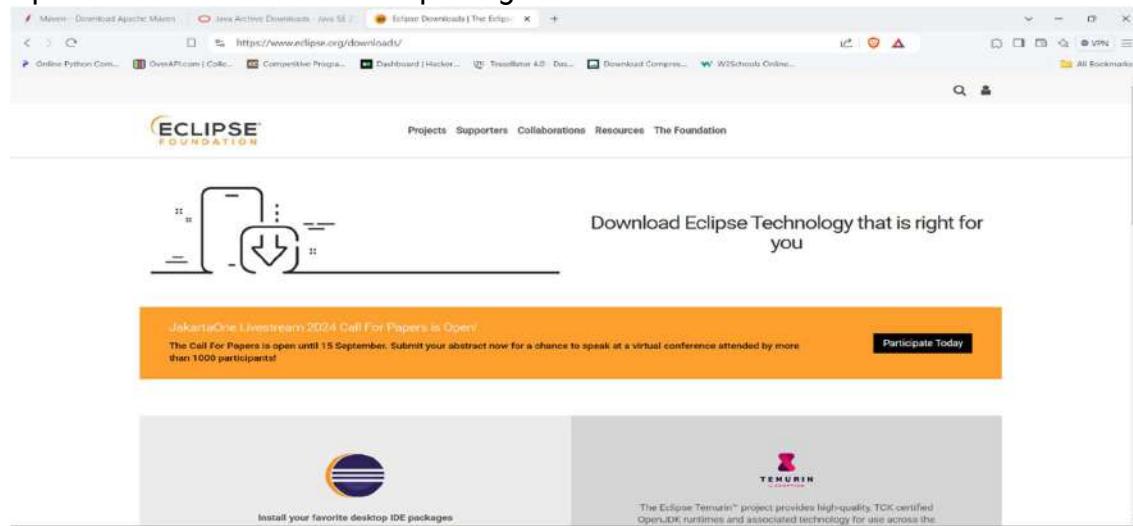


INSTALLATION OF ECLIPSE-IDE

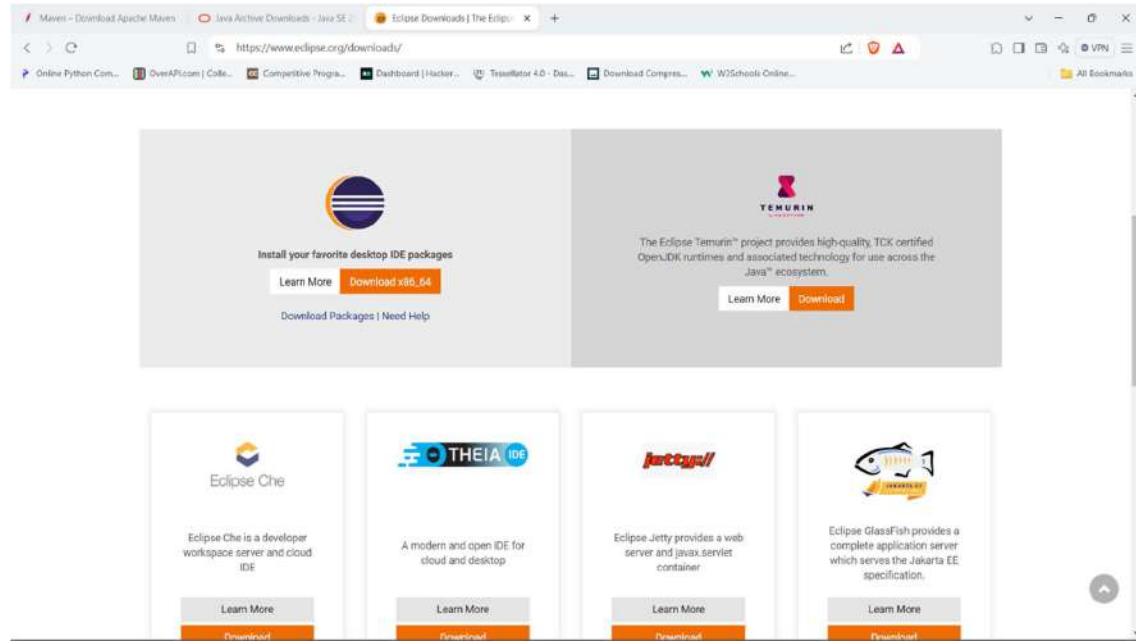
1. Go to google and search for Eclipse ide enterprise edition or eclipse ide.



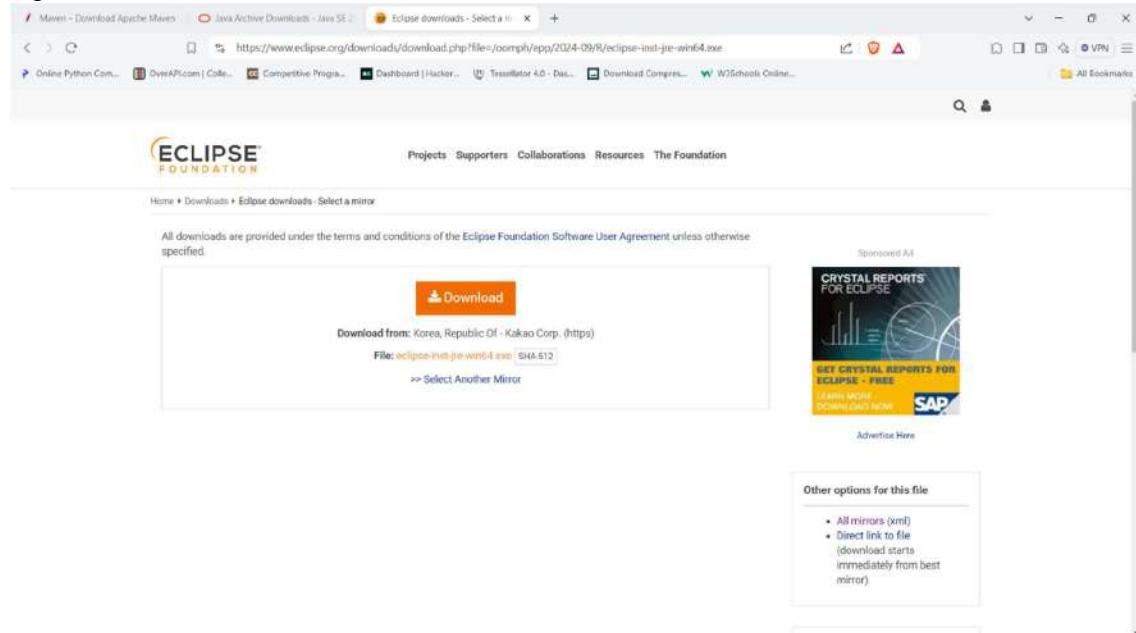
2. Open the first official link of [eclipse.org](https://www.eclipse.org/download/) .



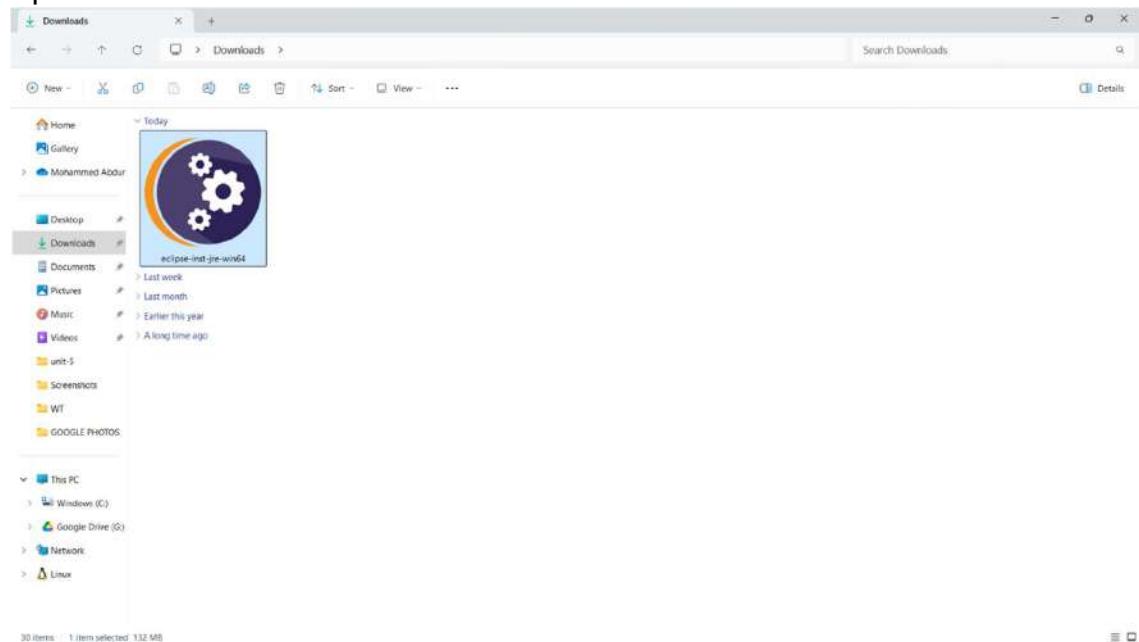
3. Now click on the download button under the eclipse ide icon(i.e Download x86_64).



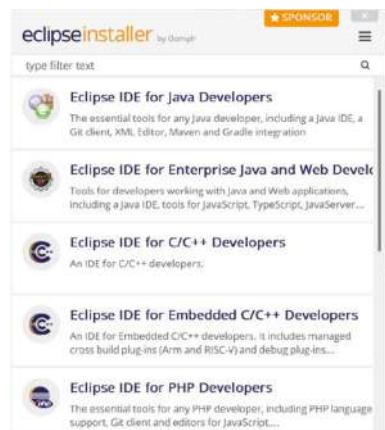
4. Again click on download.



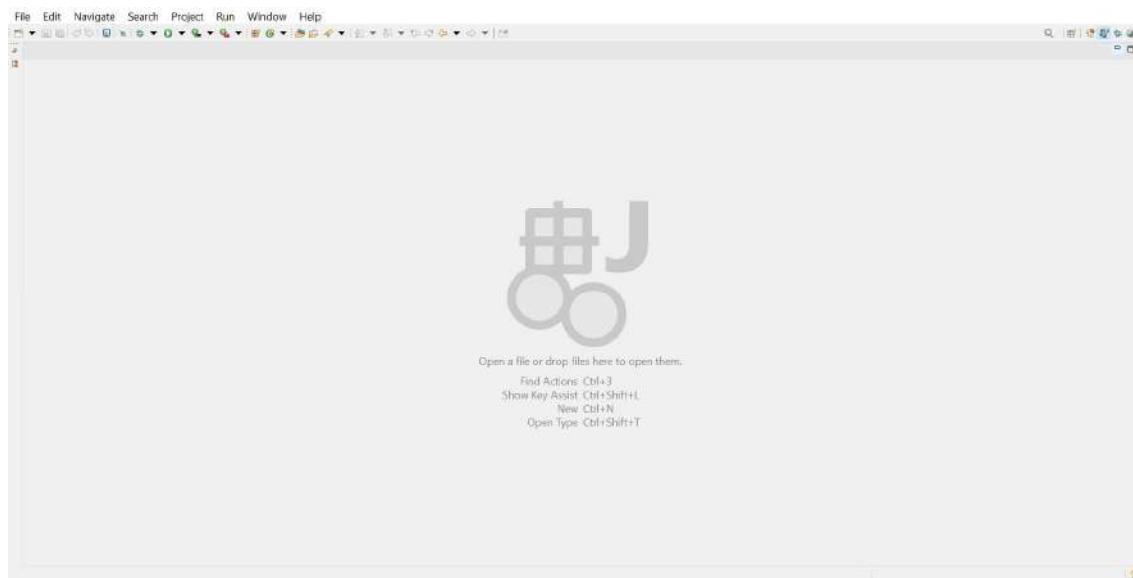
5. Open the downloads folder and run the downloaded file.



6. Select Eclipse IDE for Enterprise Java and Web Developers in the options.

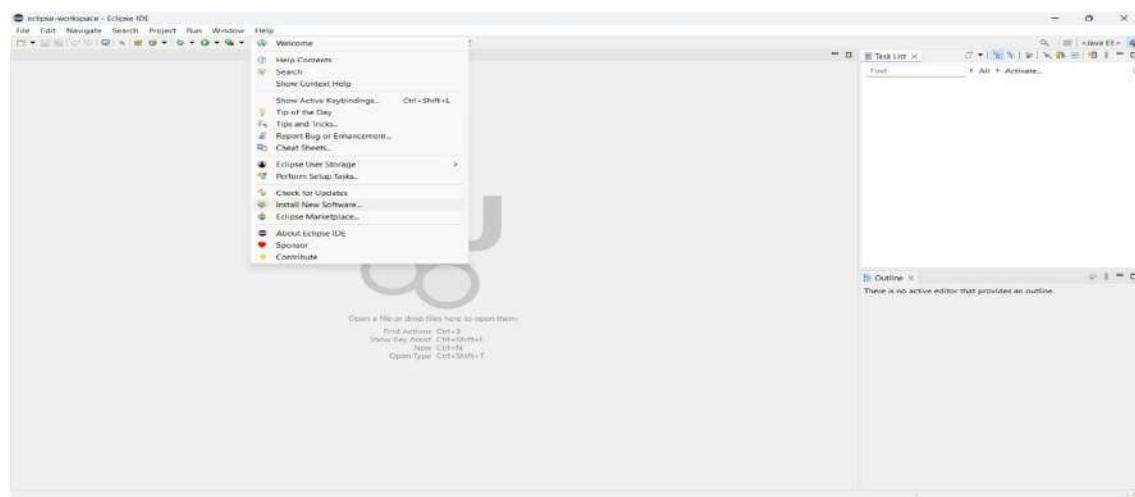


7. Interface of Eclipse IDE after successful installation.

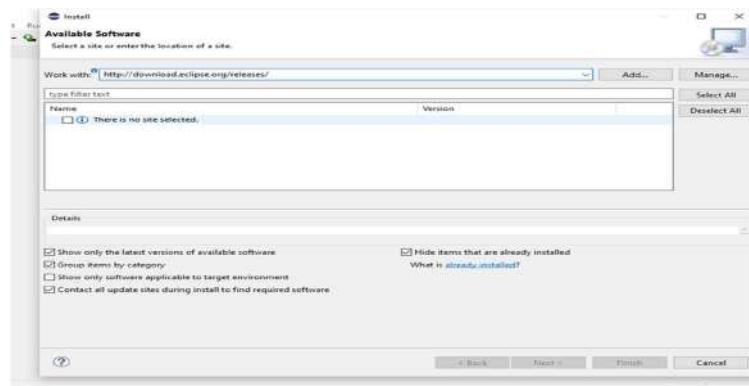


INSTALLING UPDATES INTO ECLIPSE IDE

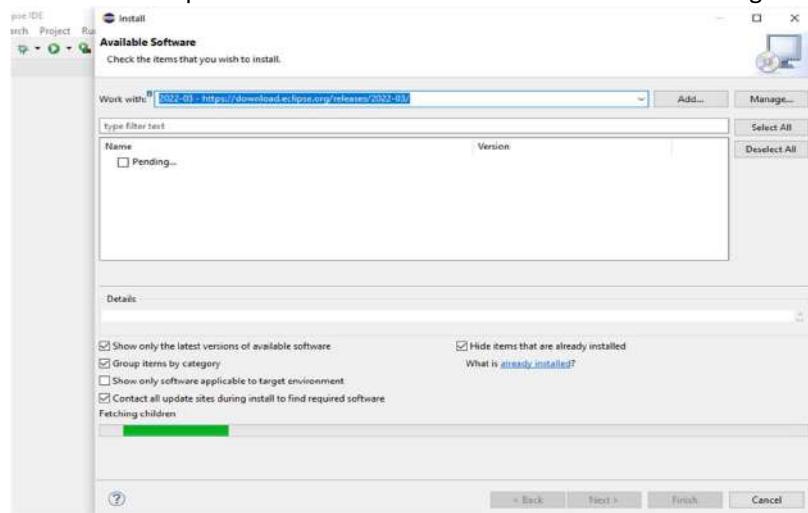
Installing updates into Eclipse



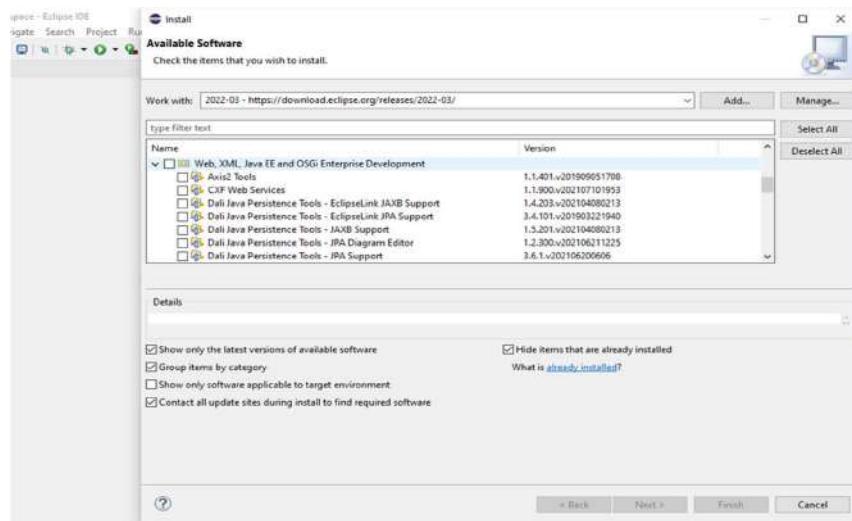
Before installing server into eclipse, we need to add Java EE to the eclipse IDE



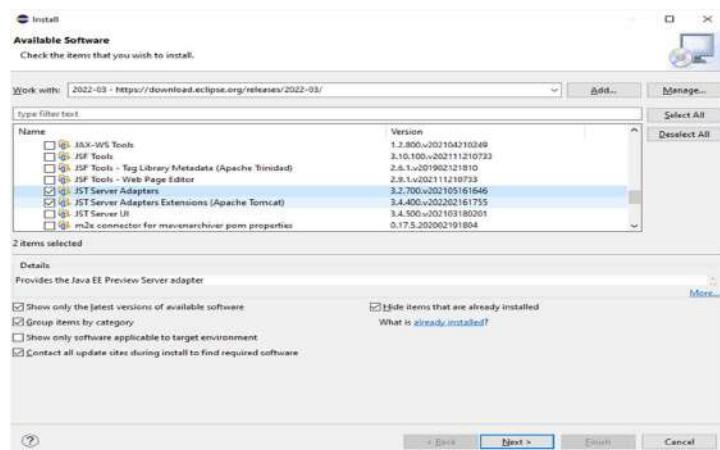
Click on the drop down menu in Work with and select the following from the list



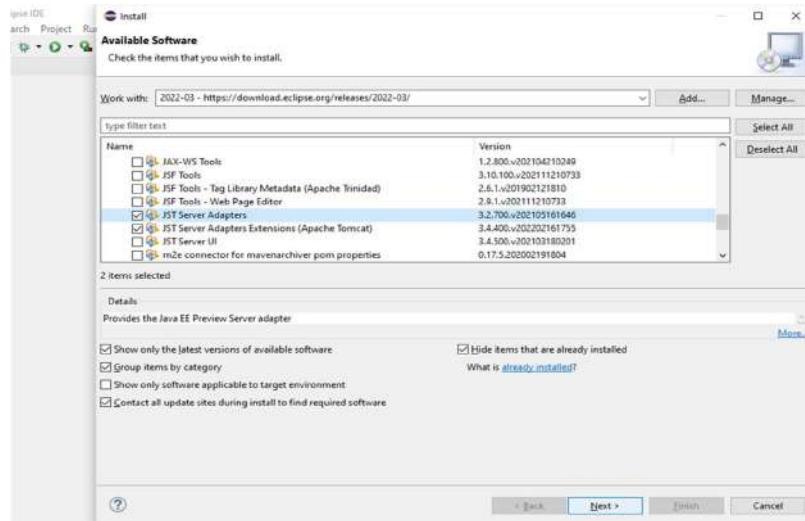
Check the Web,XML,JAVA EE box to install the latest software into eclipse in order to execute the dynamic web projects, maven web projects. Etc



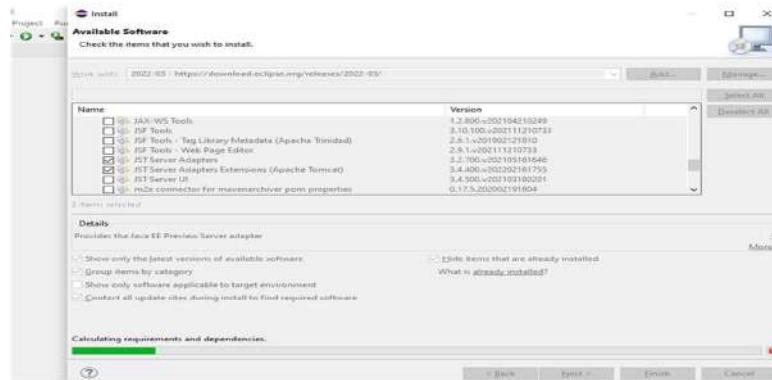
Since we develop only Maven web project in DevOps, we can only check the shown two and click on next



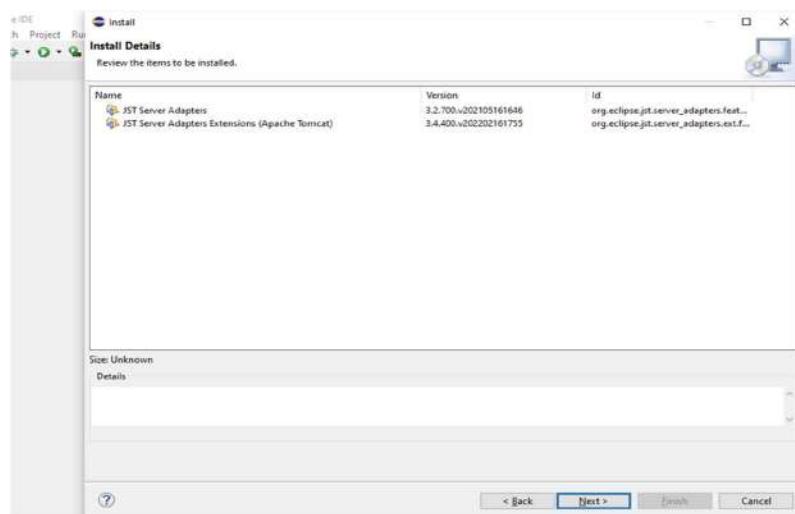
Here enable the JST Adapter and JST server adapter



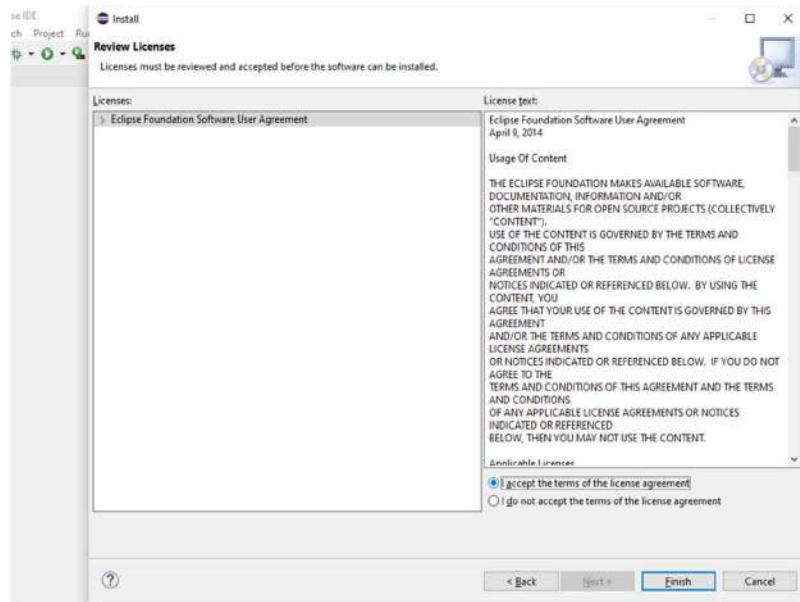
Installing Them



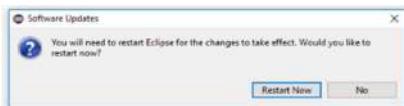
Click on -> Next



Select on -> I accept -> Finish

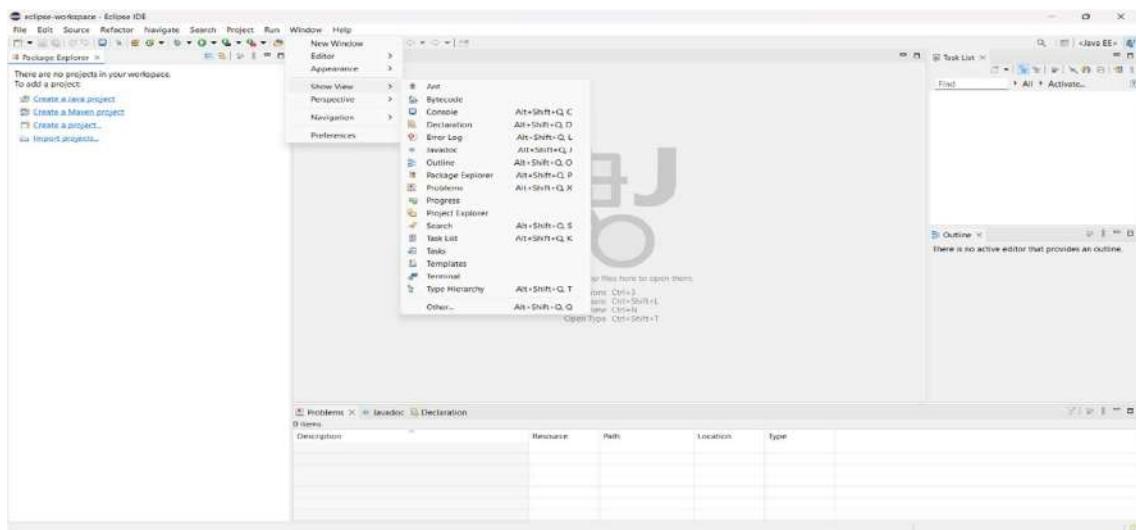


Now wait until u get Restart pop up
then → Click on Restart

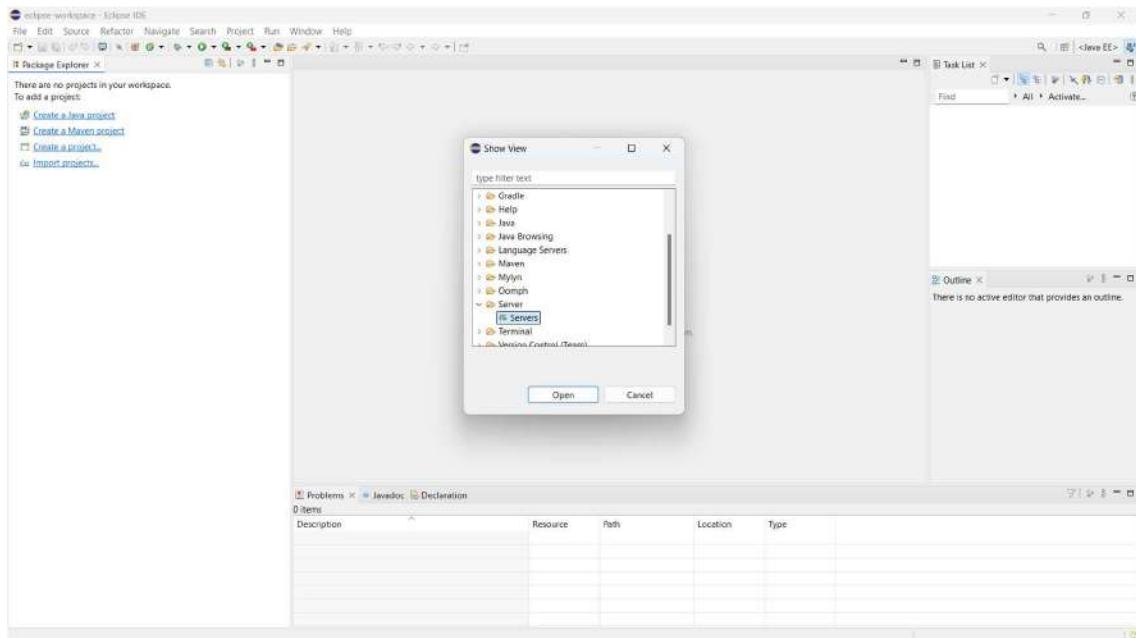


INSTALLATION OF Tomcat v9.0 in ECLIPSE-IDE

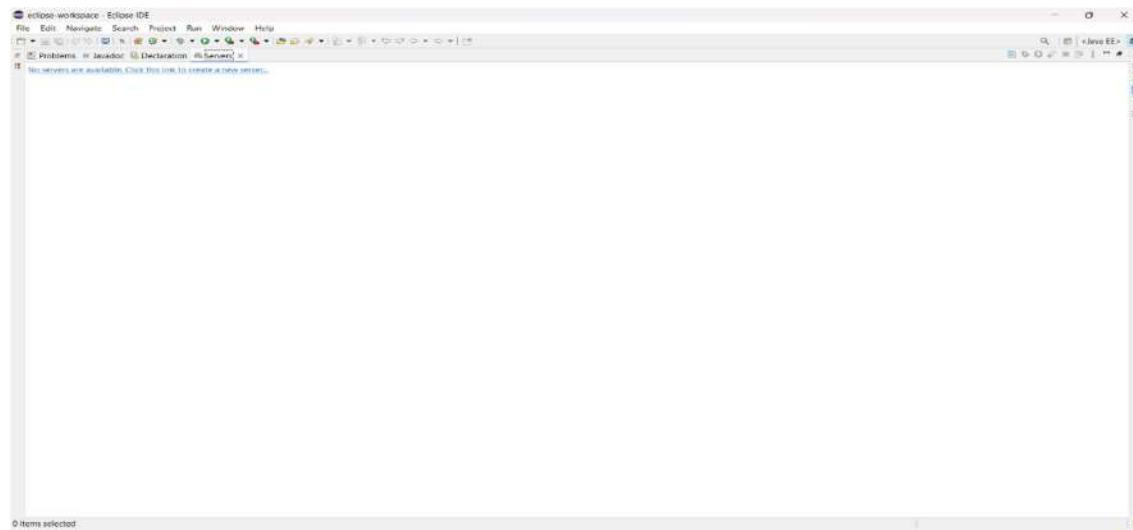
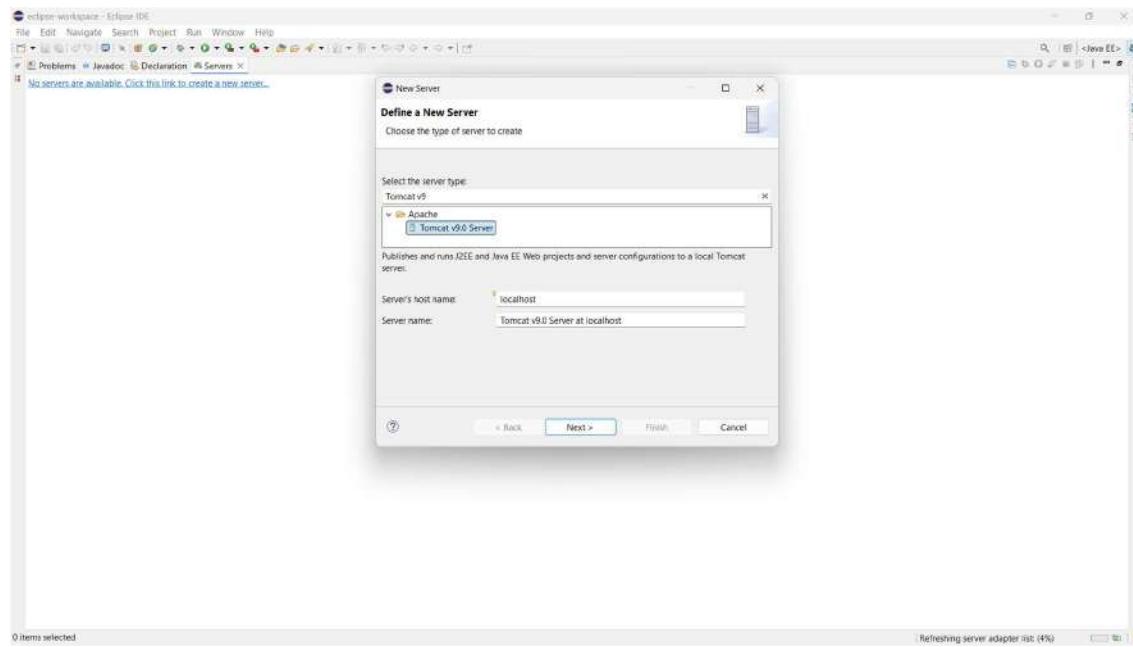
1. Go to Window in Eclipse-IDE → Show View → Other



2. Now Server→Servers



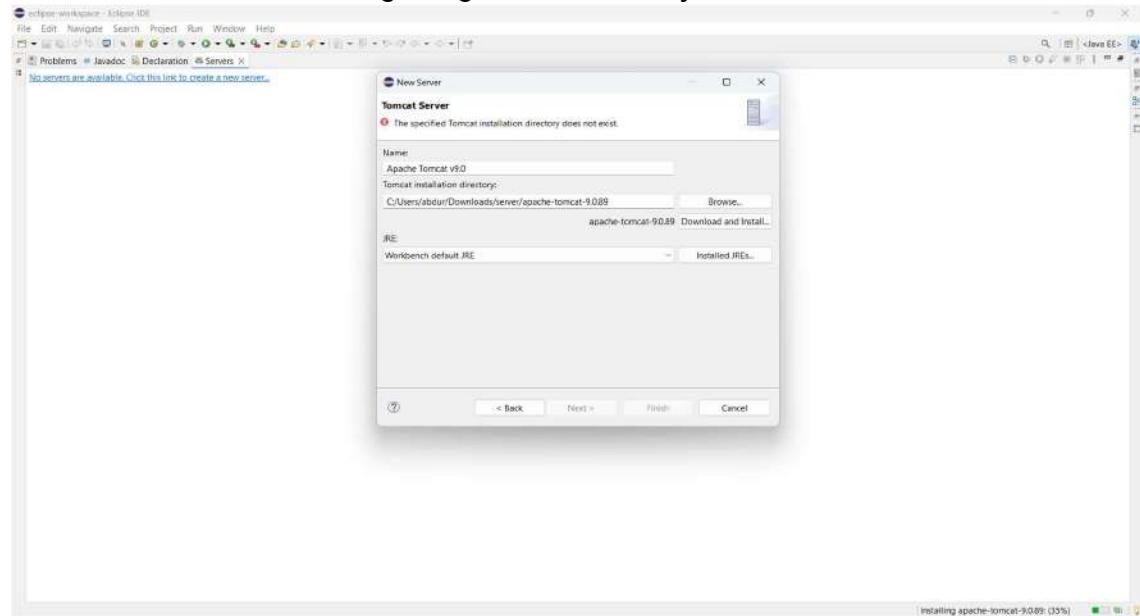
Now Open

3. Click on the link to Create new Server**4. Select Apache→Tomcat v9.0 Server
Click Next**

5. Now Download and Install → I agree... → Create an empty folder in downloads and select it

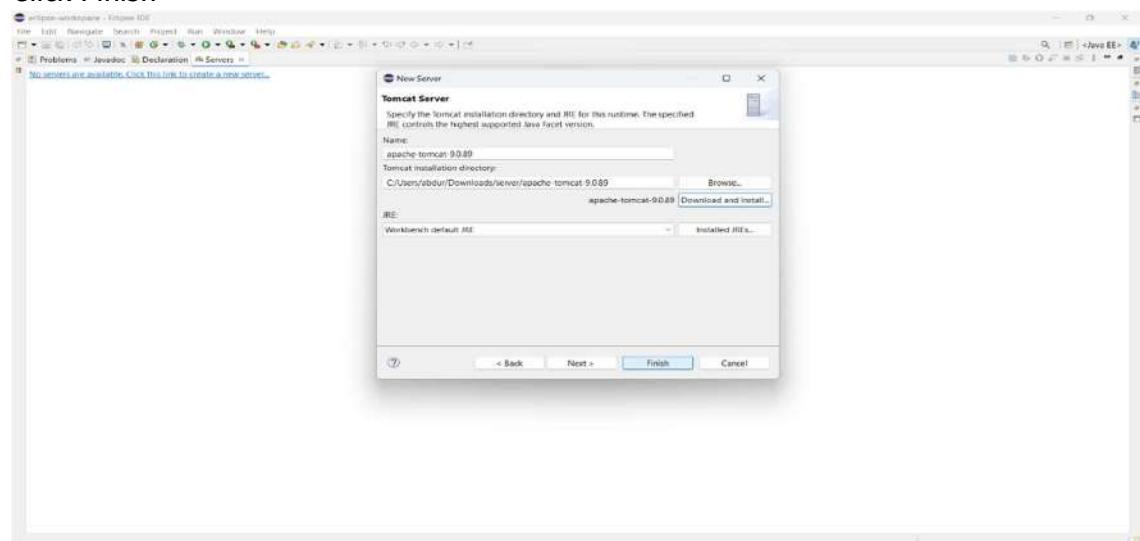
Click Download and install

as you can see in the bottom right it gets Automatically installed into that folder

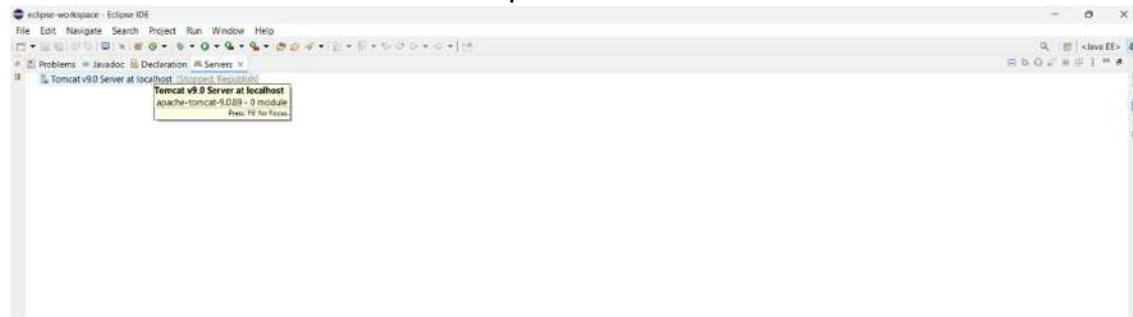


6. After the Installation is complete the Finish button is enabled

Click Finish

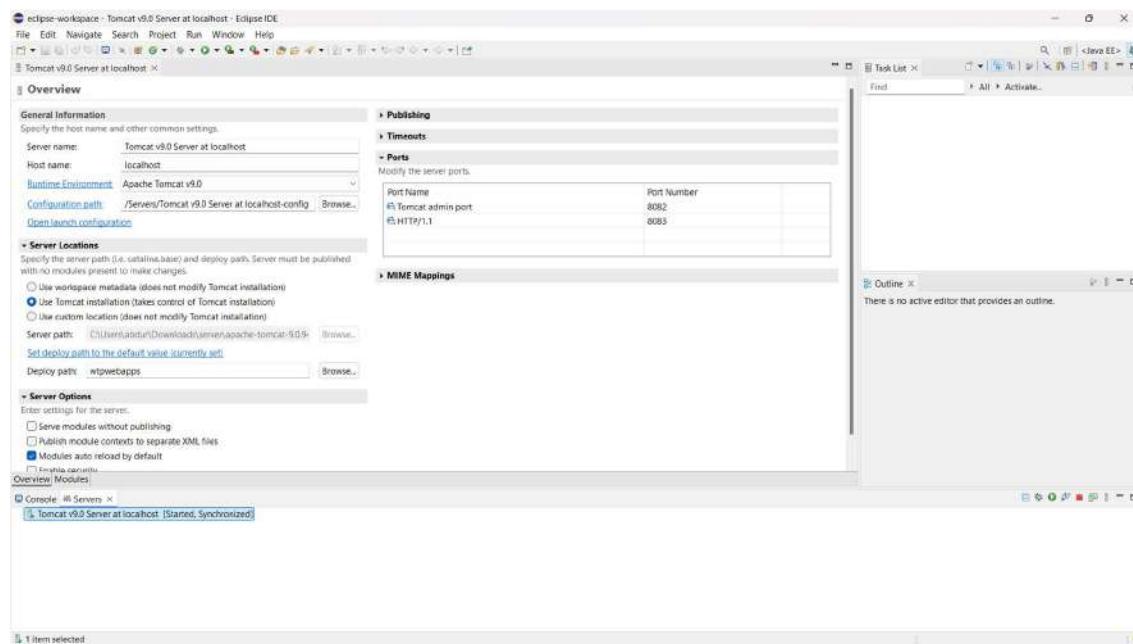


7. After installation of Tomcat v9.0 in Eclipse ide

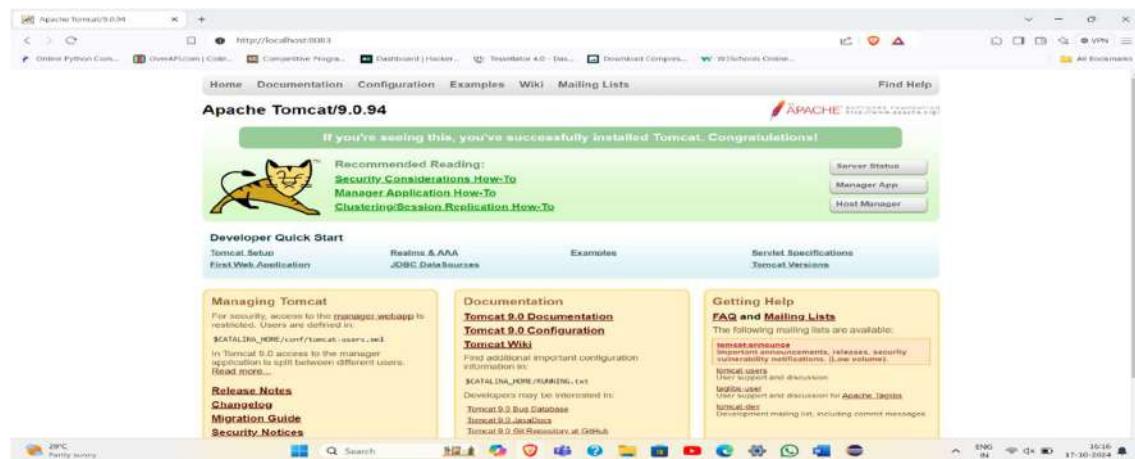


Double tap on Tomcat v9.0 Server at localhost to open Overview

8. Now change the port of Tomcat to other than 8082 as Jenkins will be using port 8080.



9. Now open the tomcat server in localhost at the changed port number



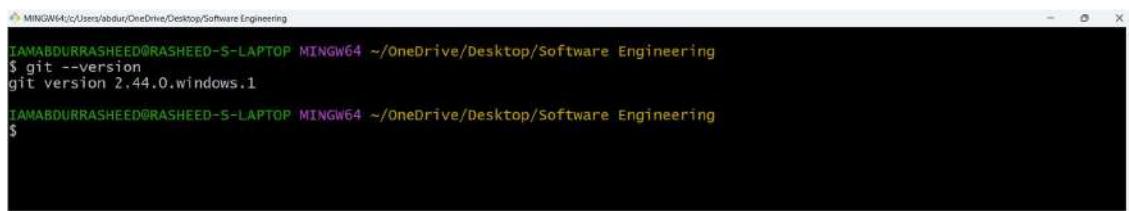
4. A. BASIC GIT COMMANDS - VERSION, CONFIG, INIT, STATUS, ADD, COMMIT, DIFF, HELP

B. GIT COMMANDS: WORKING WITH LOCAL AND REMOTE REPOSITORIES - BRANCHES, CHECKOUT, MERGE, REVERT, LOG

C. GIT COMMANDS: WORKING WITH REMOTE REPOSITORIES - REMOTE,

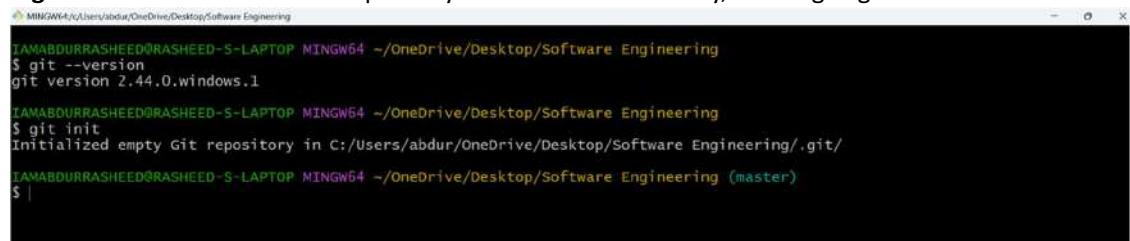
CLONE, PULL, PUSH, FORK

1.The git --version command displays the installed version of Git on your system.



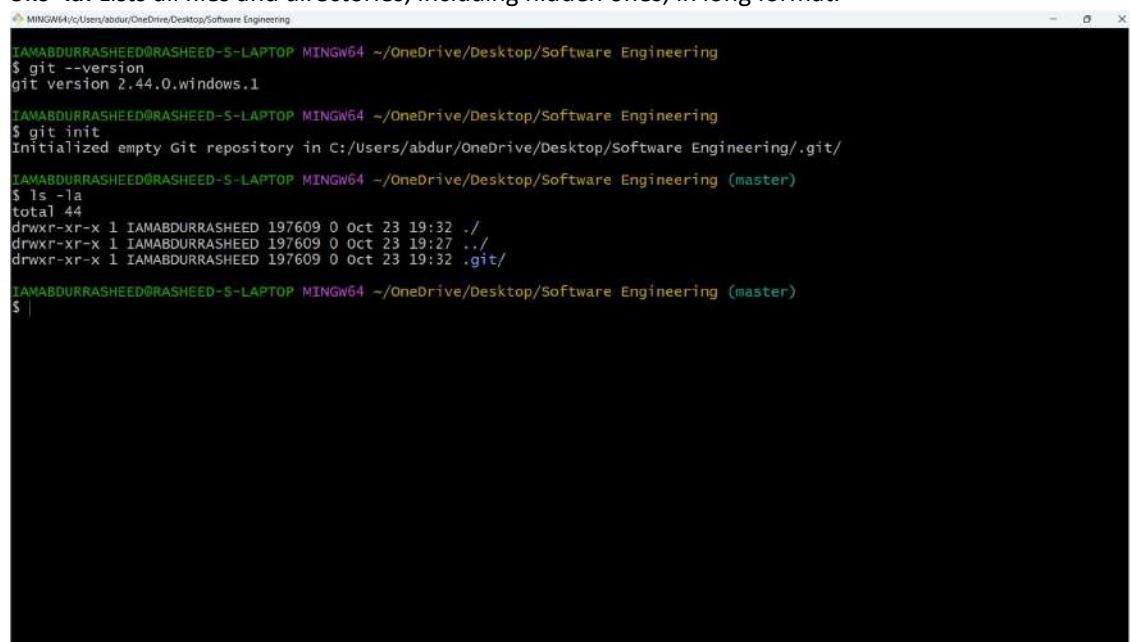
```
MINGW64 ~/OneDrive/Desktop/Software Engineering
$ git --version
git version 2.44.0.windows.1
$
```

2.git init: Initializes a new Git repository in the current directory, creating a .git folder.



```
MINGW64 ~/OneDrive/Desktop/Software Engineering
$ git --version
git version 2.44.0.windows.1
$ git init
Initialized empty Git repository in C:/Users/abdur/OneDrive/Desktop/Software Engineering/.git/
$ |
```

3.ls -la: Lists all files and directories, including hidden ones, in long format.



```
MINGW64 ~/OneDrive/Desktop/Software Engineering
$ git --version
git version 2.44.0.windows.1
$ git init
Initialized empty Git repository in C:/Users/abdur/OneDrive/Desktop/Software Engineering/.git/
$ ls -la
total 44
drwxr-xr-x 1 IAMABDURRASHEED 197609 0 Oct 23 19:32 .
drwxr-xr-x 1 IAMABDURRASHEED 197609 0 Oct 23 19:27 ../
drwxr-xr-x 1 IAMABDURRASHEED 197609 0 Oct 23 19:32 .git/
$ |
```

4.git status: Displays the state of the working directory and staging area, showing changes and untracked files.

```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ nano c.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git status
On branch master

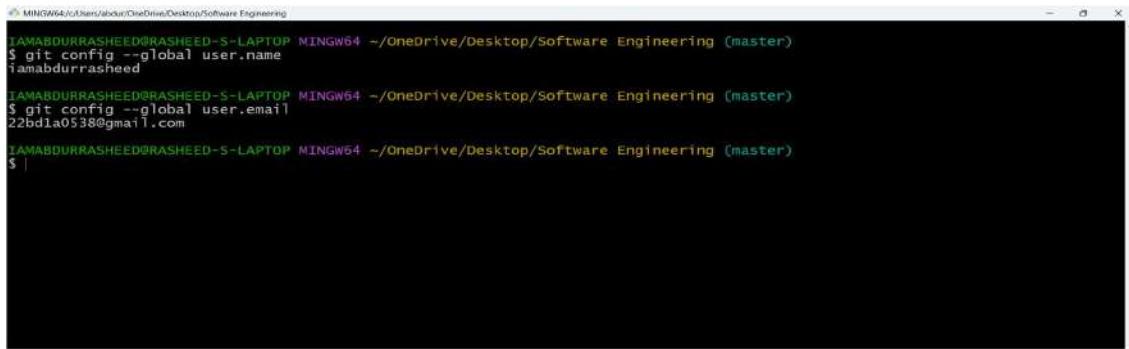
No commits yet

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

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

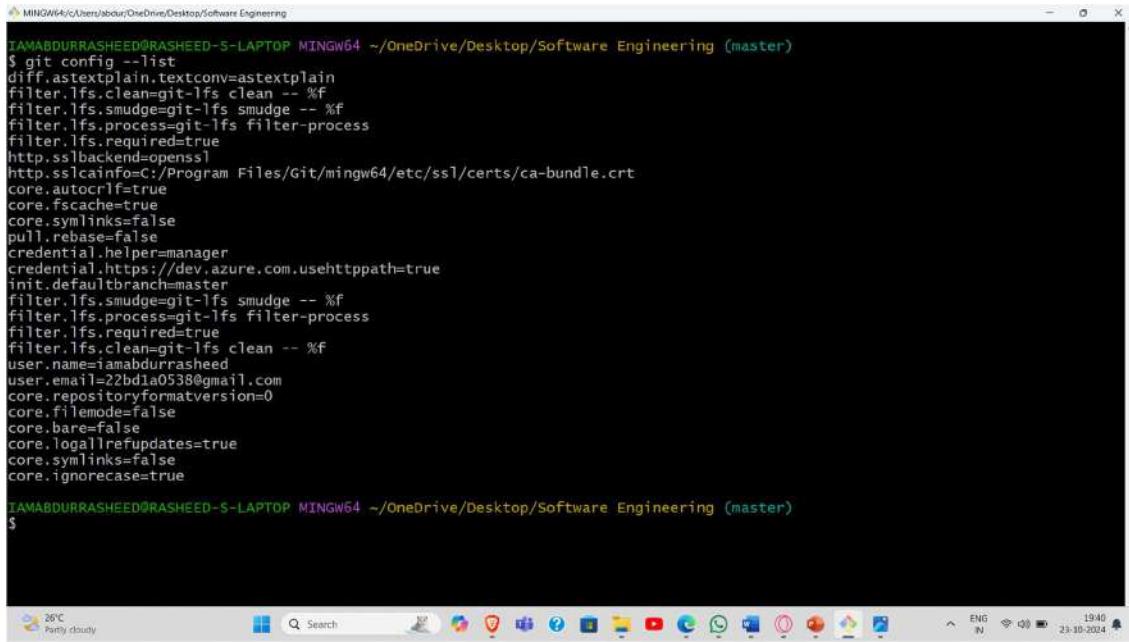
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ |
```

5.git config: Configures Git settings such as username, email, or editor preferences..



```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git config --global user.name iamabdurrasheed
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git config --global user.email 22bd1a0538@gmail.com
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ |
```

6. git config --list: Lists all the current Git configuration settings



```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (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/ssl/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
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
user.name=iamabdurrasheed
user.email=22bd1a0538@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ |
```

7. git add .: Stages all changes in the working directory, preparing them for commit.

```

TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git add .
warning: in the working copy of 'd.txt', LF will be replaced by CRLF the next time Git touches it
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git commit -m "Added new file d.txt"
[master (root-commit) b74f39a] Added new file d.txt
 2 files changed, 2 insertions(+)
 create mode 100644 c.txt
 create mode 100644 d.txt
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ 
```

8.git diff: Shows the differences between the working directory and the staging area.

```

TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ nano d.txt
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git diff
warning: in the working copy of 'd.txt', LF will be replaced by CRLF the next time Git touches it
diff --git a/d.txt b/d.txt
index ce01362..2fe1680 100644
--- a/d.txt
+++ b/d.txt
@@ -1 +1,2 @@
 hello
+hi Rasheed
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ 
```

9.git log: Displays the commit history in detail, including commit messages and authors.

```

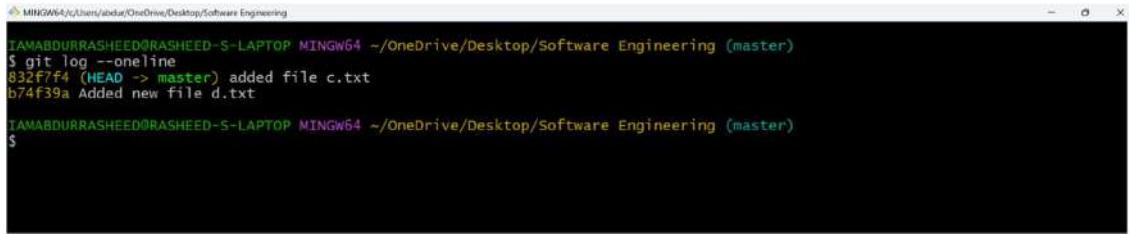
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git log
commit 832f7f412bf2544028af459f9b40990ad3a31bbc (HEAD -> master)
Author: iamabdurrasheed <22bd1a0538@gmail.com>
Date:   Wed Oct 23 19:58:43 2024 +0530

    added file c.txt

commit b74f39ad1a1624b7c01e4f7e5db8307ddff8cf963
Author: iamabdurrasheed <22bd1a0538@gmail.com>
Date:   Wed Oct 23 19:52:16 2024 +0530

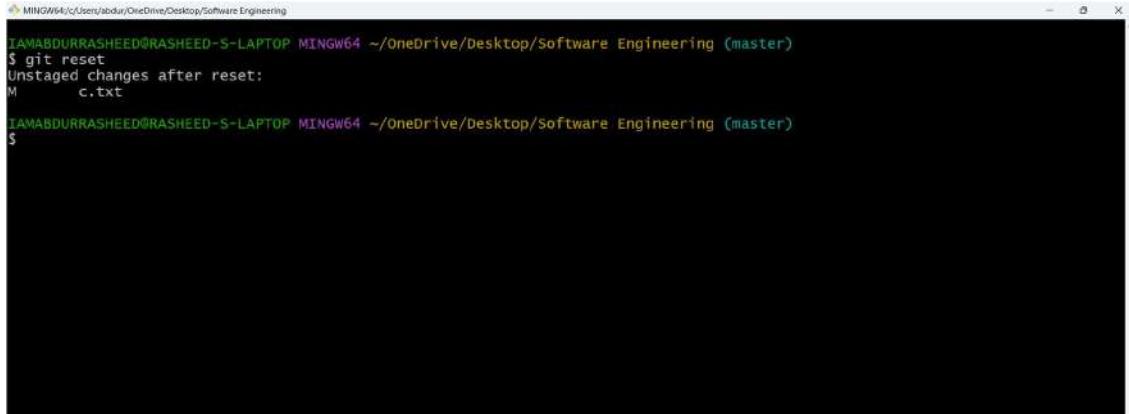
    Added new file d.txt
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ 
```

10.git log --oneline: Shows the commit history in a concise, single-line format for each commit.



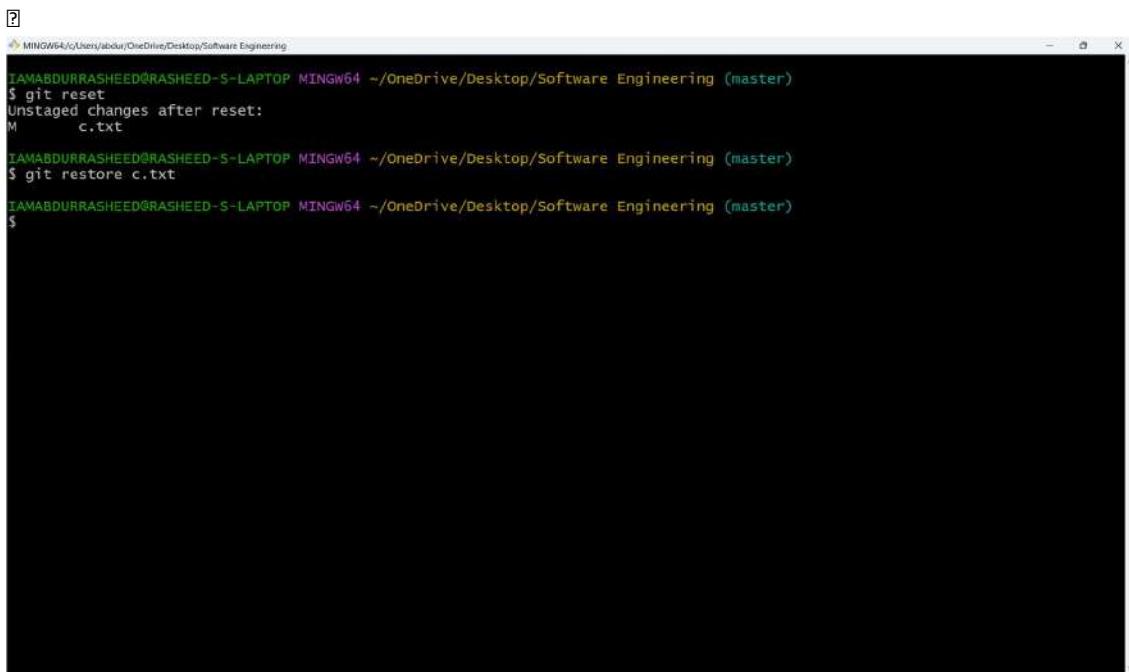
```
MINGW64/c/Users/abdur/OneDrive/Desktop/Software Engineering
$ git log --oneline
832f7f4 (HEAD -> master) added file c.txt
b74f39a Added new file d.txt
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$
```

11.git reset: Unstages changes or resets commits, modifying the commit history.



```
MINGW64/c/Users/abdur/OneDrive/Desktop/Software Engineering
$ git reset
Unstaged changes after reset:
M     c.txt
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$
```

12.② git restore: Restores files in the working directory to a previous state or discards changes.



```
MINGW64/c/Users/abdur/OneDrive/Desktop/Software Engineering
$ git reset
Unstaged changes after reset:
M     c.txt
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git restore c.txt
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$
```

13.Git diff after reset .

```
MINGW64:/c/Users/abdul/OneDrive/Desktop/Software Engineering
$ git diff

```

The terminal shows the command \$ git diff, which outputs nothing as there are no changes.

14.git branch: Lists, creates, or deletes branches in the repository.

```
MINGW64:/c/Users/abdul/OneDrive/Desktop/Software Engineering
$ git branch project_changes

```

```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git branch -a
* master
  project_changes

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git branch -d project_changes
Deleted branch project_changes (was 832f7f4).

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git branch -a
* master

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$
```

The terminal shows the creation of a new branch 'project_changes' and its subsequent deletion.

15.git checkout: Switches between branches or restores files to a specific state.

```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering
$ git branch -a
* master
  project_changes

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git checkout project_changes
Switched to branch 'project_changes'

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_changes)
$ ls
c.txt d.txt

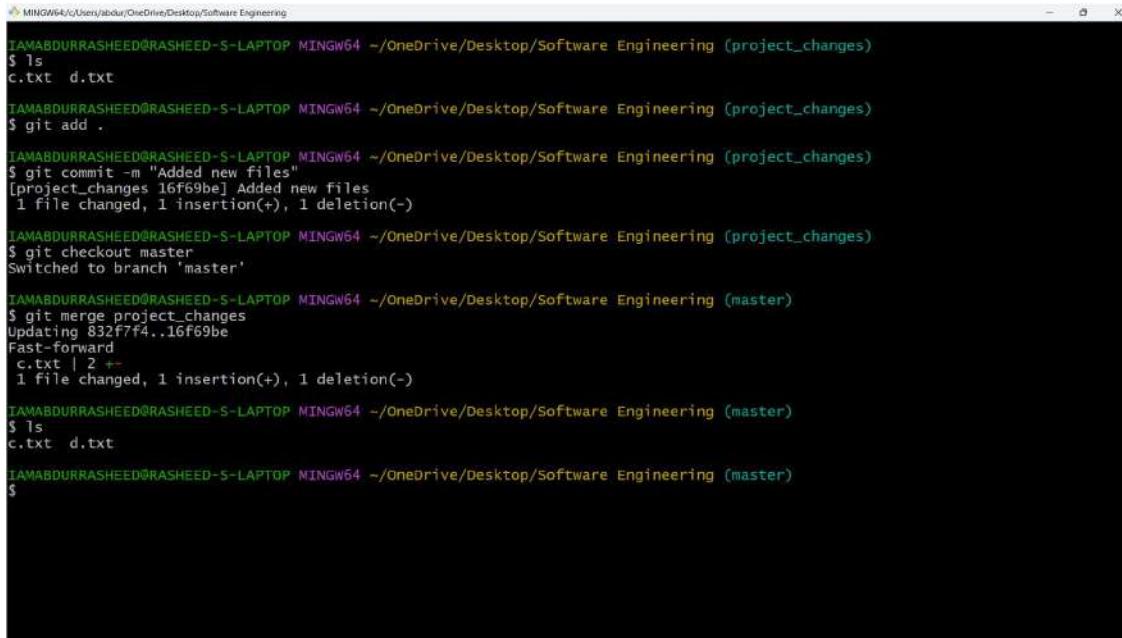
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_changes)
$ git checkout -b project_bugs
Switched to a new branch 'project_bugs'

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_bugs)
$ git branch -a
* project_bugs
  master
  project_changes

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_bugs)
$
```

The terminal shows the switching between branches 'master', 'project_changes', and 'project_bugs'. It also lists the contents of the 'project_bugs' directory, which contains files 'c.txt' and 'd.txt'.

16.git merge: Combines changes from different branches into the current branch.



```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_changes)
$ ls
c.txt d.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_changes)
$ git add .

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_changes)
$ git commit -m "Added new files"
[project_changes 16f69be] Added new files
 1 file changed, 1 insertion(+), 1 deletion(-)

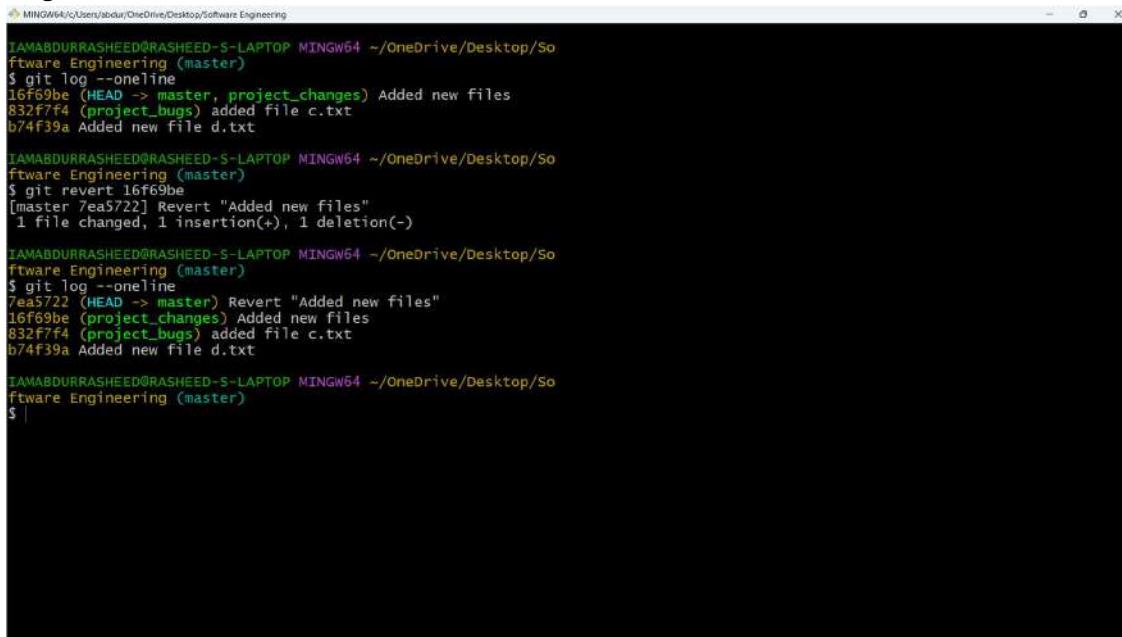
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (project_changes)
$ git checkout master
Switched to branch 'master'

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git merge project_changes
Updating 832f7f4..16f69be
Fast-forward
 c.txt | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ ls
c.txt d.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$
```

17.git revert: Creates a new commit that undoes



```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git log --oneline
16f69be (HEAD -> master, project_changes) Added new files
832f7f4 (project_bugs) added file c.txt
b74f39a Added new file d.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git revert 16f69be
[master 7ea5722] Revert "Added new files"
 1 file changed, 1 insertion(+), 1 deletion(-)

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git log --oneline
7ea5722 (HEAD -> master) Revert "Added new files"
16f69be (project_changes) Added new files
832f7f4 (project_bugs) added file c.txt
b74f39a Added new file d.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ |
```

Git revert opens the below editor

```

MINGW64/c/Users/abdur/Desktop/ogg
$ nano c.txt
ZAHABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/ogg (project_changes)
$ git add c.txt
ZAHABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/ogg (project_changes)
warning: in the working copy of 'c.txt', LF will be replaced by CRLF the next time Git touches it
ZAHABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/ogg (project_changes)
$ git commit -m "committed"
[project_changes 0a39070] committed
 2 files changed, 2 insertions(+)
  create mode 100644 b.txt
  create mode 100644 c.txt
ZAHABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/ogg (project_changes)
$ git log --oneline
0a39070 (HEAD => project_changes) committed
0243c2c (project_bugs, master) Create new file
ZAHABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/ogg (project_changes)
$ git revert 0a39070
[project_changes 6c3b69d] Revert "committed"
 2 files changed, 2 deletions(-)
 delete mode 100644 b.txt
 delete mode 100644 c.txt
ZAHABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/ogg (project_changes)
$
```

To exit the Git commit message editor, press Esc, type :wq, and hit Enter.

```

MINGW64/c/Users/abdur/Desktop/ogg
Revert "committed"

This reverts commit 0a39070f66b887ee45aabe7b9238df0ad3f66e58.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch project_changes
# Changes to be committed:
#   deleted:  b.txt
#   deleted:  c.txt
#
#
.git/COMMIT_EDITMSG [unix] (12:00 25/10/2024) 1,1 A11
:wq
```

**.Git help Command:

```

$ git help
usage: git [-v] [--version] [-h] [--help] [-c <path>] [-c <name>=<value>]
        [-c <key_id>=<value>] [-c <key_id>[<path>]] [-c <key_id>[<path>]]
        [-p] [--paginate] [-p] [--no-pager] [--no-replace-objects] [--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 copy a file, 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
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

```

Git help merge redirected to this page

git-merge - Join two or more development histories together

SYNOPSIS

```

git merge [-n] [-stat] [--no-commit] [--squash] [--[no-]edit]
          [--no-verify] [-s <strategy>] [-X <strategy-option>] [-S <key_id>]
          [--[no-]allow-unrelated-histories]
          [--[no-]rerere-autoupdate] [-m <msg>] [-F <file>]
          [--into-name <branch>] [<commit>...]
git merge (--continue | --abort | --quit)

```

DESCRIPTION

Incorporates changes from the named commits (since the time their histories diverged from the current branch) into the current branch. This command is used by **git pull** to incorporate changes from another repository and can be used by hand to merge changes from one branch into another.

Github Dashboard

The screenshot shows the GitHub dashboard with the following elements:

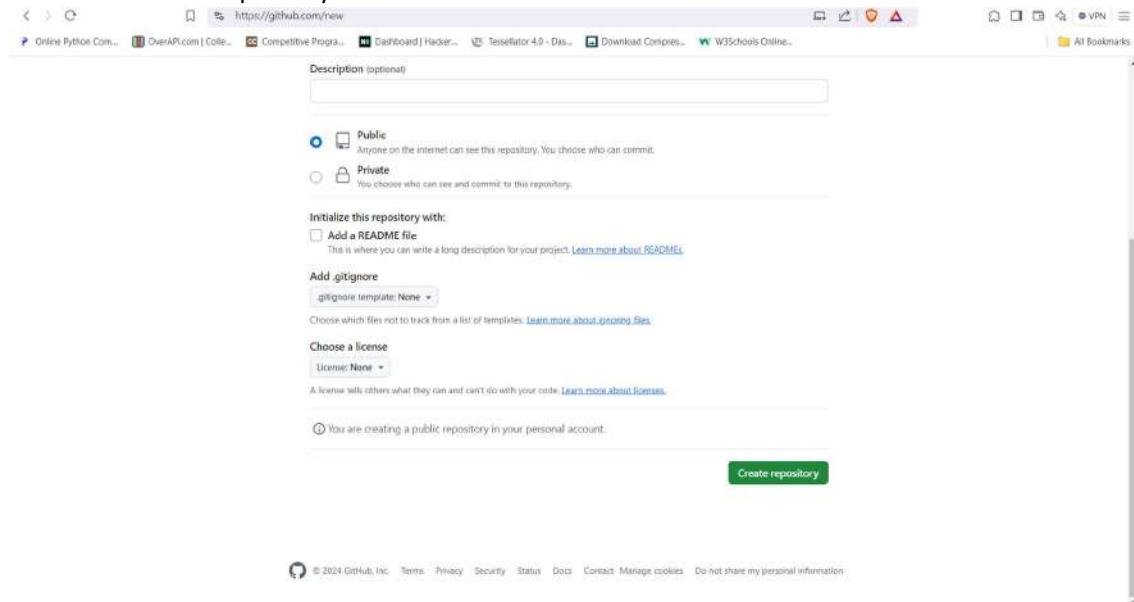
- Top repositories:** Shows repositories like `iamabdurrashheed/iamabdurrashheed`, `RohitCodes/twink`, and `divyaranjan12599/gym-data-management`.
- Start writing code:** A button to begin coding.
- Create a new repository:** A form where you can enter a repository name (e.g., `sample_example`) and choose between Public or Private settings. The Public option is selected.
- Repositories that need your help:** A section listing projects like `gradle / gradle` and `joaomatosilva / DateTimeExtensions`.
- UNIVERSE'24:** An advertisement for the event with a "Get tickets" button.
- Latest changes:** A feed of recent updates from GitHub, including bypass controls for push protection and enterprise scanning improvements.
- Explore repositories:** A link to browse more repositories.

19. Creating a new repository in GitHub.

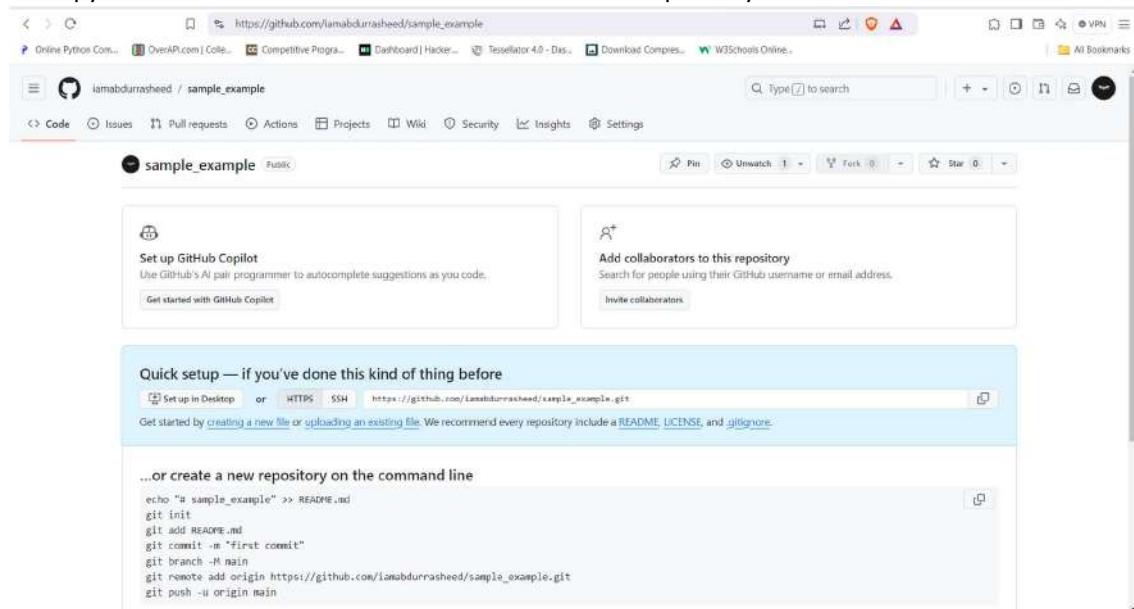
The screenshot shows the "Create a new repository" form with the following fields filled in:

- Owner:** `iamabdurrashheed`
- Repository name:** `sample_example`
- Description (optional):** A placeholder text: "Great repository names are short and memorable. Need inspiration? How about: redesigned-spoon ?"
- Visibility:** The Public radio button is selected.
- Initialize this repository with:** The "Add a README file" checkbox is checked.
- Add .gitignore:** A dropdown menu set to "None".
- Choose which files not to track:** A link to learn more about ignoring files.

20.Click on Create repository.



21.Copy the HTTPS URL Link to connect with the Github Repository.



22.Git remote command to establish a connection between git and Github.

```

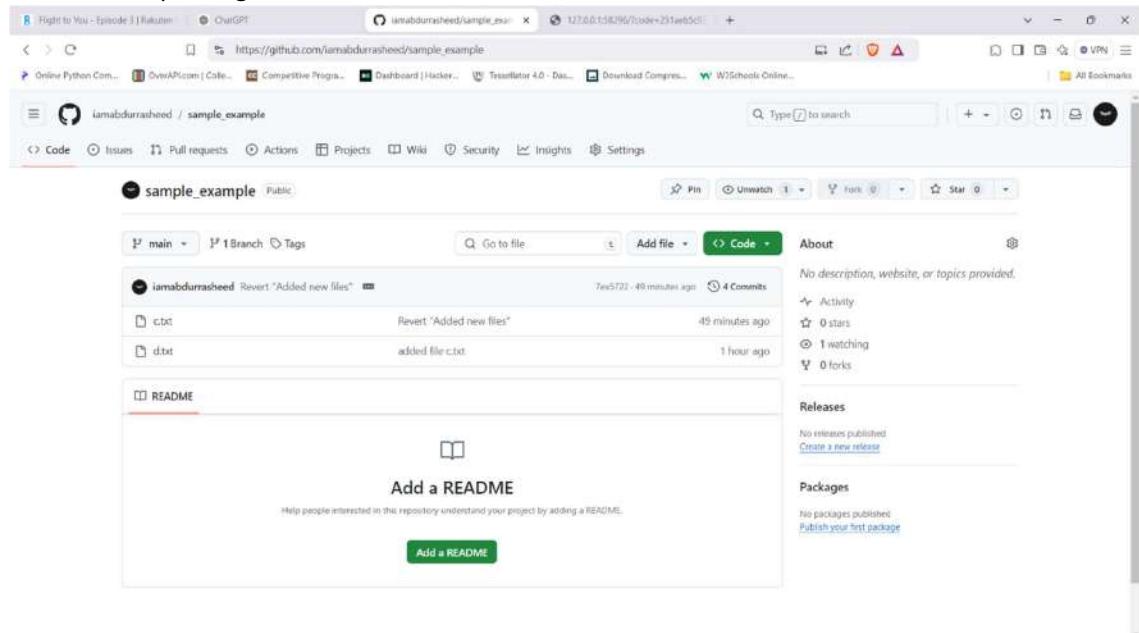
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git remote add origin https://github.com/iamabdurrasheed/sample_example.git
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git remote -v
origin https://github.com/iamabdurrasheed/sample_example.git (fetch)
origin https://github.com/iamabdurrasheed/sample_example.git (push)
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ 
```

23.Switch to main branch and push the files.

```

TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (master)
$ git branch -M main
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 947 bytes | 315.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/iamabdurrasheed/sample_example.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ 
```

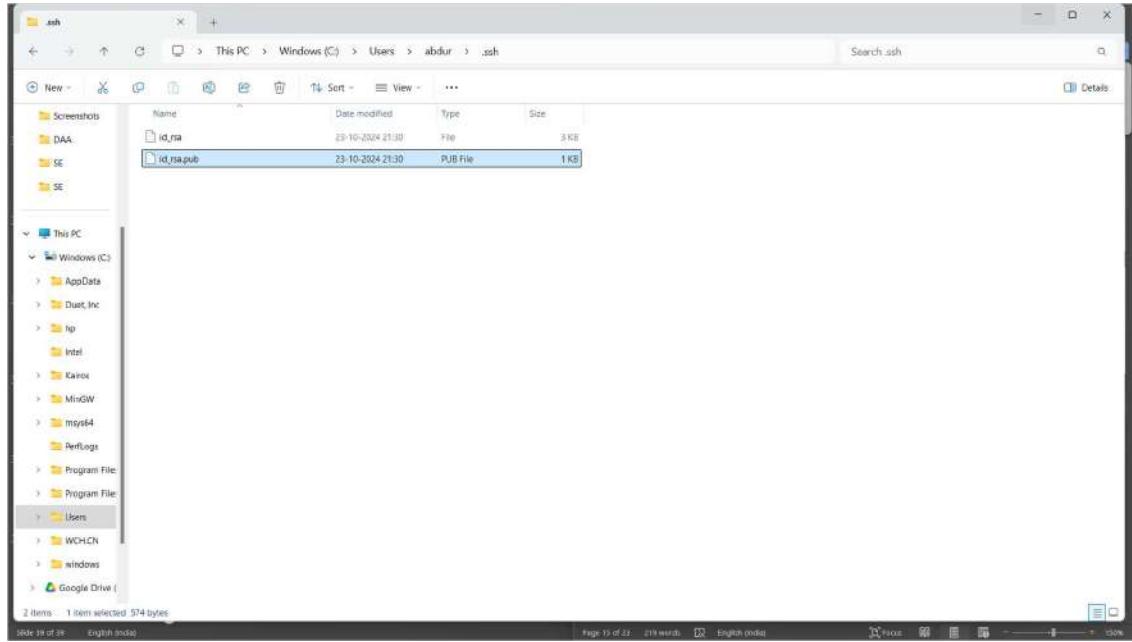
24. Files after pushing to Github.



25. Generating SSH key using (ssh-keygen -t rsa -C 22bd1a0538@gmail.com).

```
MINGW64 ~/OneDrive/Desktop/Software Engineering
$ ssh-keygen -t rsa -C 22bd1a0538@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/abdur/.ssh/id_rsa)
a):
Created directory '/c/Users/abdur/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/abdur/.ssh/id_rsa
Your public key has been saved in /c/Users/abdur/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ksDDbyU9cBekhne8JwdLBZm9ssK2NiW2cjXteDP/oYc 22bd1a0538@gmail.com
The key's randomart image is:
+---[RSA 3072]---+
|          .+..   |
|          ..%    |
|          ..* + . |
| + o .+.* . .  |
| B o o+So . .  |
| = ..=+ + .    |
| + oo=*= =,    |
| o .B+E...     |
| .+.+oo.       |
+---[SHA256]---+
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$
```

26.SSH key stored at /c/Users/abdur/.ssh/id_rsa.pub



27.The SSH Key that was generated.

```

id_rsa.pub
File Edit View
-----BEGIN RSA PUBLIC KEY-----[REDACTED]
-----END RSA PUBLIC KEY-----[REDACTED]
-----BEGIN RSA PRIVATE KEY-----[REDACTED]
-----END RSA PRIVATE KEY-----[REDACTED]

```

In 2, Col 1 374 of 374 characters

28.Goto Github and navigate to SSH and GPG and add new SSH key.

Mohammed Abdur Rasheed (iamabdurrasheed)
Your personal account

Add new SSH Key

Title:

Key type: Authentication Key

Key:

```
ssh-rsa AAAQAB3NzC1yc2EAAAQABgQC...  
AAAQAB3NzC1yc2EAAAQABgQC...7Qoekh...omPRICjH.../u...mWgGE0V...CBL.../4k+HeU...Pr...5dQJPF...4R53a...heDTkW1...rRGd5mGKMSKIC  
yo...9t...dA...ZT1...lB...m...2...M...P...Y...M...N...0...X...k...k...C...S...X...1...9...0...T...V...c...1...4...u...u...S...C...D...2...R...V...B...9...c...+...Y...X...u...u...M...S...O...I...3...C...L...X...Y...v...74...M.../...4...C...7...C...V...9...1...h...5...B...Z...p...f...3...c...S...d...P...L...N...  
Or...B...M.../...H...m...t...D...W...W...E...m...k...p...4...e...j...n...u...s...t...l...X...6...p...j...Z...n...T...T...r...4...l...J...D...P...p...y...G...W...S...T...A...t...+...B...V...x...g...z...e...d...7...A...V...W.../...d...X...y...d...k...+...y...C...R...N...t...1...R...0...3...l...4...z...r...e...m...c...4...v...k...l...z...W...f...P...U...P...6...F...1...1...D...o...u...K...3...  
A...6...2...3...1...+...A...E...m...y...h...C...Y...7...a...b...E...f...L...u...v...X...g...5...k...D...4...v...b...0...4...C...M...e...X...n...l...O...s...+...7...2...w...w...l...N...5...N...+...d...7...3...M...P...k...p...0...E...6...6.../...S...d...5...h...w...q...N...Q...Q...W...5...6...4...6...Z...a...t...C...k...y...f...T...U...5...h...q...G...K...Y...B...R...b...  
d...5...s.../...0...c...5...j...q...n...e...m...v...0...d...0...7...y...7...F...1...h...6.../...m...D...k...q...d...5...p...1...7...B...N...Q...q...D...e...m...d...s...7...Z...u...w...P...p...V...z...v...o...Z...5...0...e...0...c...+...c...=...2...2...b...d...1...a...0...5...3...@...g...m...a...i...l...c...o...m...
```

Add SSH key

29.SSH key successfully added.

You have successfully added the key '22bd1a0538@gmail.com'.

Mohammed Abdur Rasheed (iamabdurrasheed)
Your personal account

SSH keys

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

Authentication keys

22bd1a0538@gmail.com	5h425...+...d...0...d...f...b...k...h...8...h...E...Z...v...e...s...t...2...h...1...Q...C...5...h...e...D...P.../...Y...c... SSH	Added on Oct. 23, 2024	Delete
----------------------	---	------------------------	------------------------

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

New GPG key

Vigilant mode

30.Cloning a Repository copy the url of SSH you want to clone

This repository has been archived by the owner on Oct 26, 2018. It is now read-only.

Clone

HTTPS SSH GitHub CLI

git@github.com:orf/simple.git

Use a password-protected SSH key.

Local Codespaces

Simple is a clone of Obtuse written in Python running on Flask.

README MIT license

Activity 506 stars 27 watching 203 forks Report repository

No releases published

No packages published

31.Git clone command.

```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ git clone git@github.com:orf/simple.git
Cloning into 'simple'...
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3vvv6TuJhbPzisF/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: 1436, done.
remote: Total 1436 (delta 0), reused 0 (delta 0), pack-reused 1436 (from 1)
Receiving objects: 100% (1436/1436), 2.34 MiB | 92.00 KiB/s, done.
Resolving deltas: 100% (739/739), done.

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ ls -ltr
total 6
-rw-r--r-- 1 IAMABDURRASHEED 197609 17 Oct 23 19:53 d.txt
-rw-r--r-- 1 IAMABDURRASHEED 197609 22 Oct 23 20:35 c.txt
drwxr-xr-x 1 IAMABDURRASHEED 197609 0 Oct 23 21:43 simple/
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$
```

32.Making changes to the Github repository.

Name	Last commit message	Last commit date
a.txt	Create a bit	2 minutes ago
c.txt	Revert "Added new files"	1 hour ago
d.txt	added file c.txt	1 hour ago

33.Git pull command

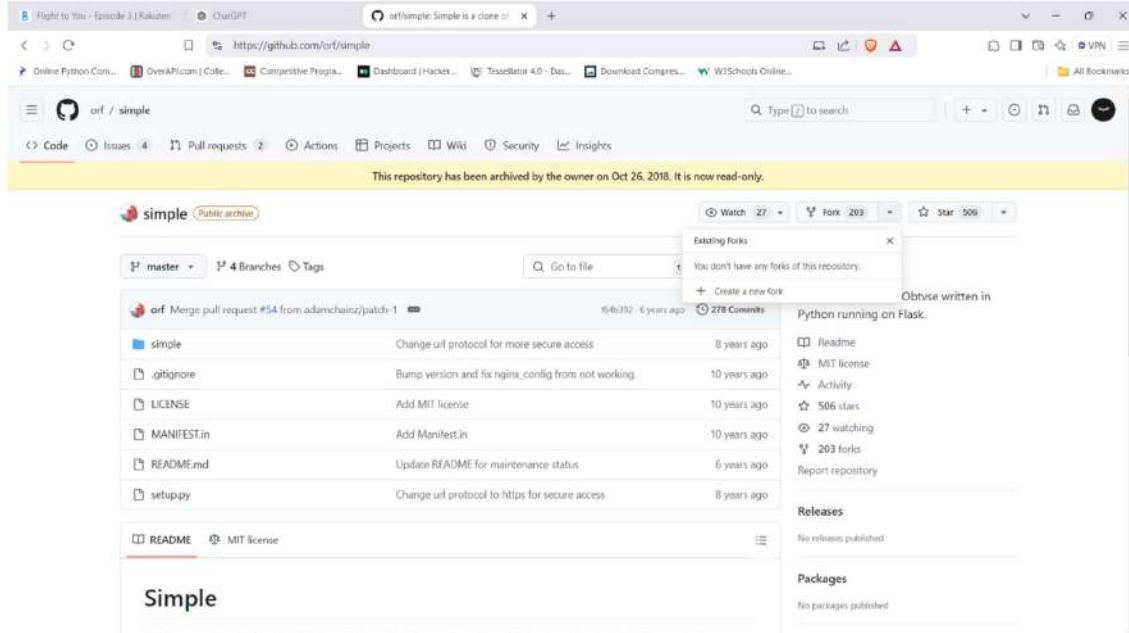
```

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ ls -ltr
total 6
-rw-r--r-- 1 IAMABDURRASHEED 197609 17 Oct 23 19:53 d.txt
-rw-r--r-- 1 IAMABDURRASHEED 197609 22 Oct 23 20:35 c.txt
drwxr-xr-x 1 IAMABDURRASHEED 197609 0 Oct 23 21:43 simple/
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 960 bytes | 60.00 KiB/s, done.
From https://github.com/iamabdurrasheed/simple_example
    7ea5722..2d71667  main      -> origin/main
Updating 7ea5722..2d71667
Fast-forward
 a.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 a.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ ls -ltr
total 7
-rw-r--r-- 1 IAMABDURRASHEED 197609 17 Oct 23 19:53 d.txt
-rw-r--r-- 1 IAMABDURRASHEED 197609 22 Oct 23 20:35 c.txt
drwxr-xr-x 1 IAMABDURRASHEED 197609 0 Oct 23 21:43 simple/
-rw-r--r-- 1 IAMABDURRASHEED 197609 11 Oct 23 21:49 a.txt

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/Software Engineering (main)
$ 
```

34.Forking a repository. Open a new public repository.Click on Fork on right side



35. Give a Repository name

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 * **Repository name ***

iamabdurnasheed / simple

simple is 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 Obtuse by adding your own branch. [Learn more](#).

ⓘ You are creating a fork in your personal account.

Create fork

Click on Create fork.

36. Repository is forked on your GitHub account

simple Public

forked from [ort/simple](#)

master • 1 Branch Tags

This branch is up to date with [ort/simple/master](#).

Commits

Author	Commit Message	Date	Commits
ort	Merge pull request #54 from adamchainz/patch-1	104b292 - 6 years ago	278 Commits
simple	Change url protocol for more secure access	8 years ago	
.gitignore	Bump version and fix nginx_config from not working	10 years ago	
LICENSE	Add MIT license	10 years ago	
MANIFEST.in	Add Manifest.in	10 years ago	
README.md	Update README for maintenance status	6 years ago	
setup.py	Change url protocol to https for secure access	8 years ago	

About

Simple is a clone of Obtuse written in Python running on Flask.

- [Readme](#)
- [MIT license](#)
- [Activity](#)
- [0 stars](#)
- [0 watching](#)
- [0 forks](#)

Releases

No releases published.

[Create a new release](#)

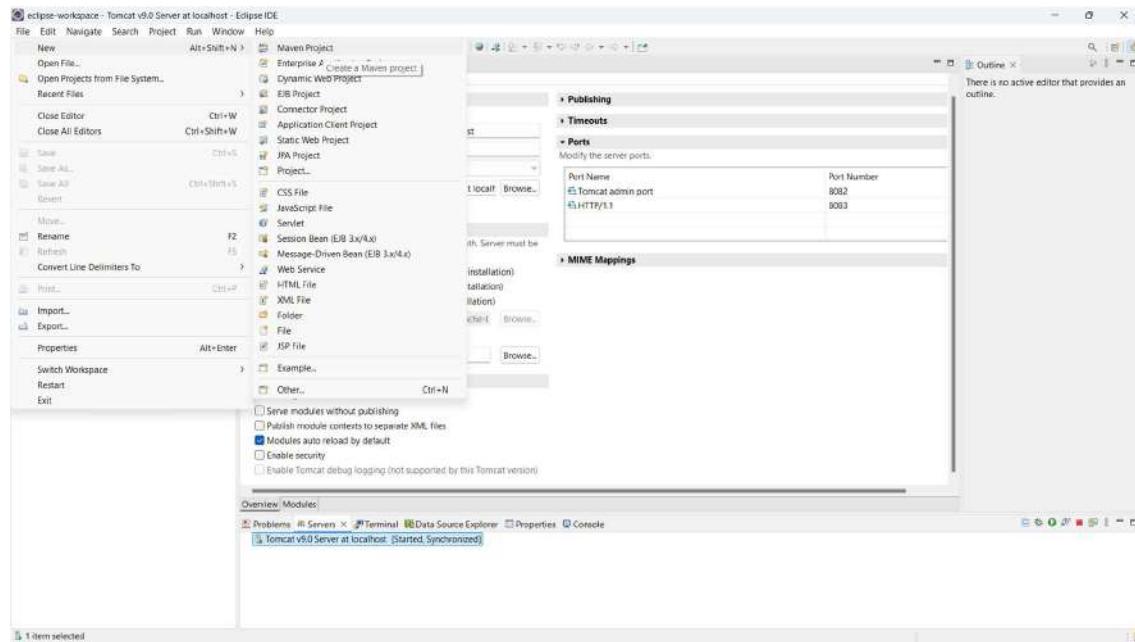
Packages

No packages published.

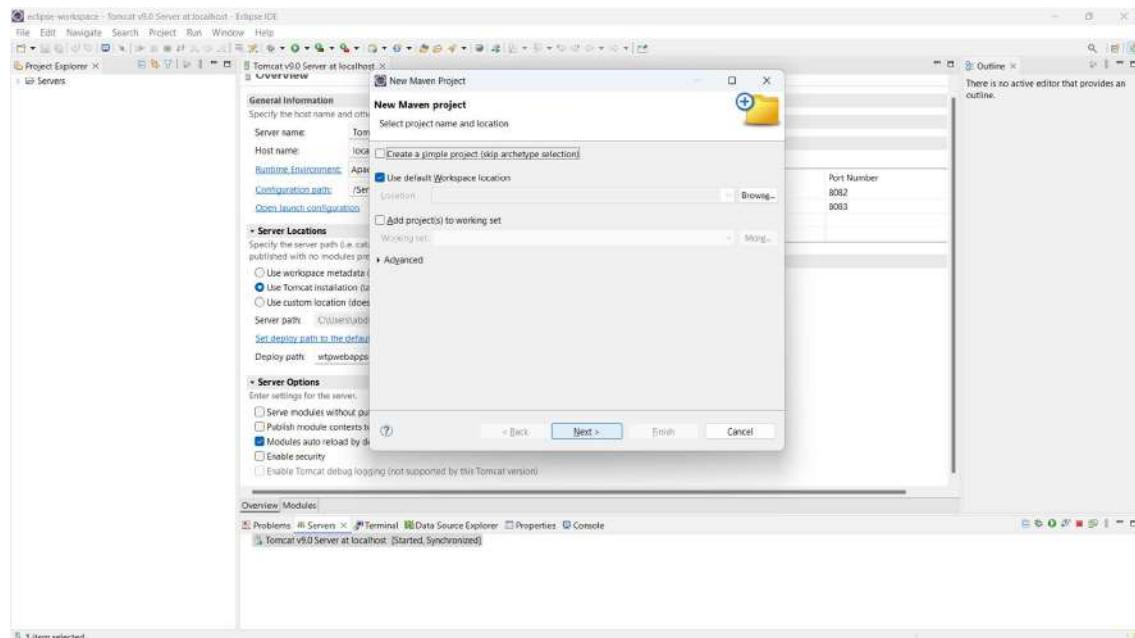
[Publish your first package](#)

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

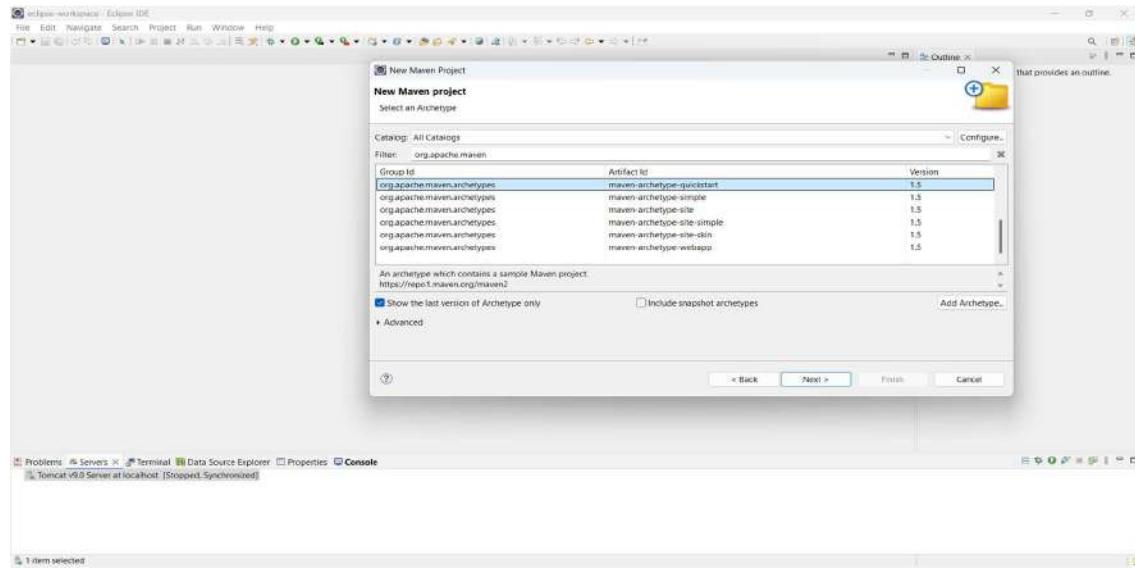
Create a Maven Project - > File -> New -> Maven Project



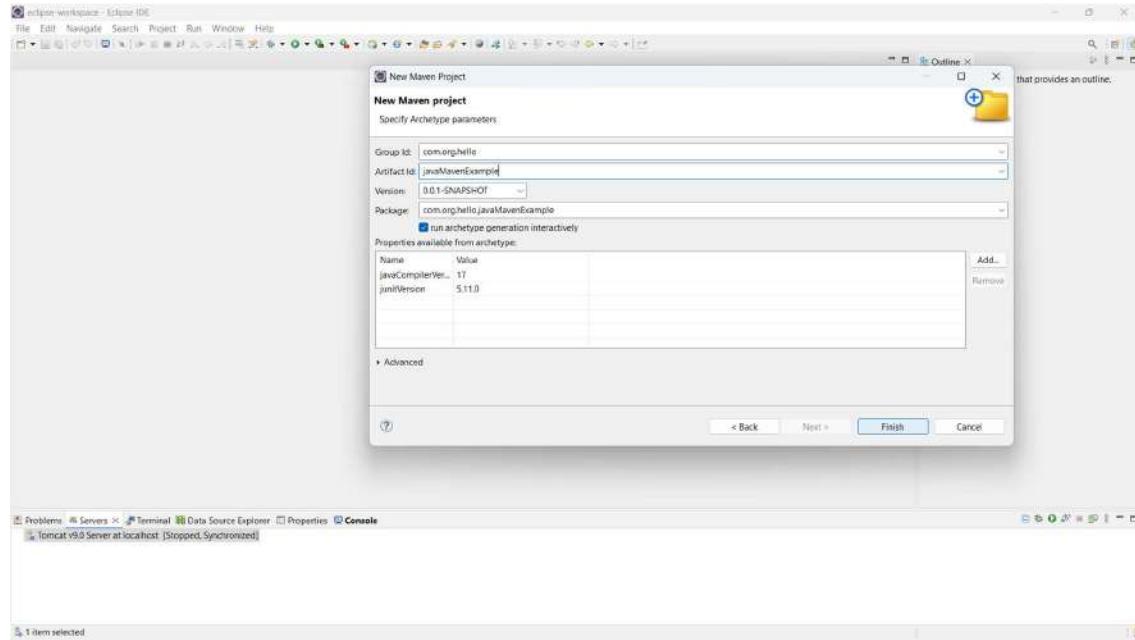
Click on Next



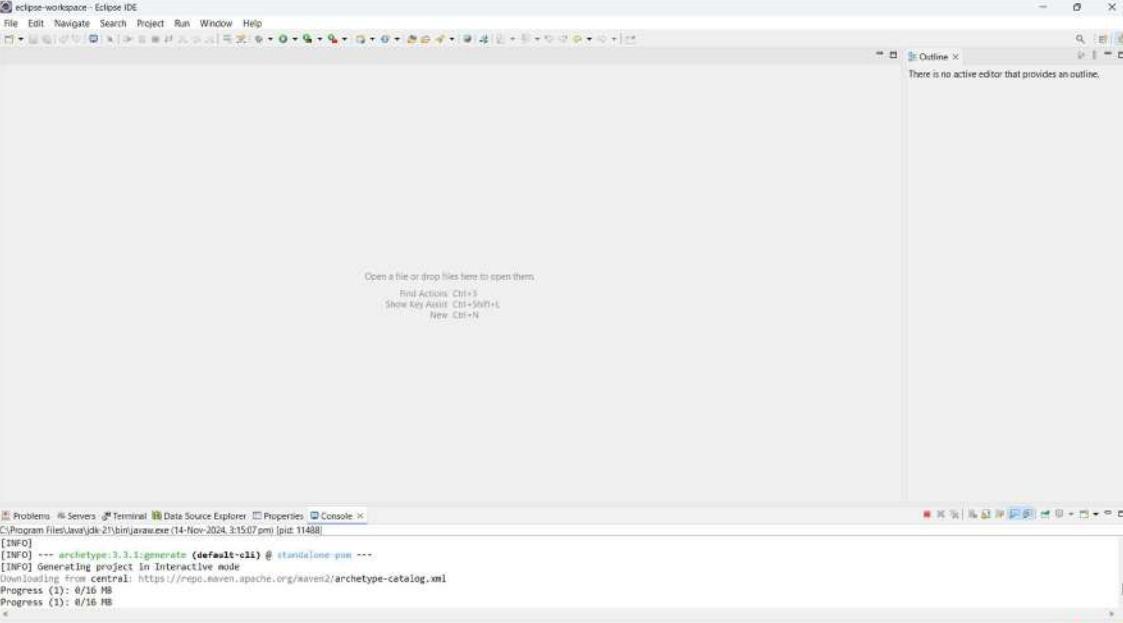
Select the quickstart archetype under org.apache.maven.archetypes and click on Next



Give Artifact ID of your choice and click Finish, Maven Project is created



Wait till y: occurs in the console

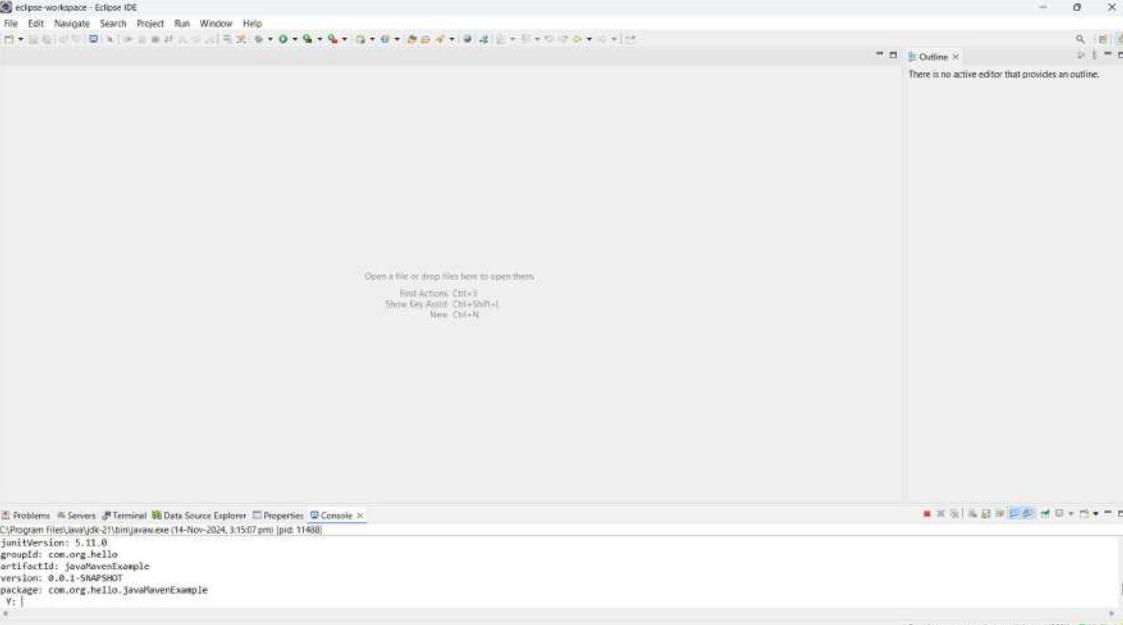


The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - Eclipse IDE". The central area displays the Maven command-line interface (CLI) output in the "Console" tab:

```
[INFO] [INFO] -- archetype: 0.3.1:generate {default-cli} @ standalone-pom ***
[INFO] [INFO] Generating project in interactive mode
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml
Progress (1): 0/16 MB
Progress (1): 0/16 MB
y
```

The status bar at the bottom right indicates "Creating maven-archetype-quickstart (33%)".

Press enter next to y

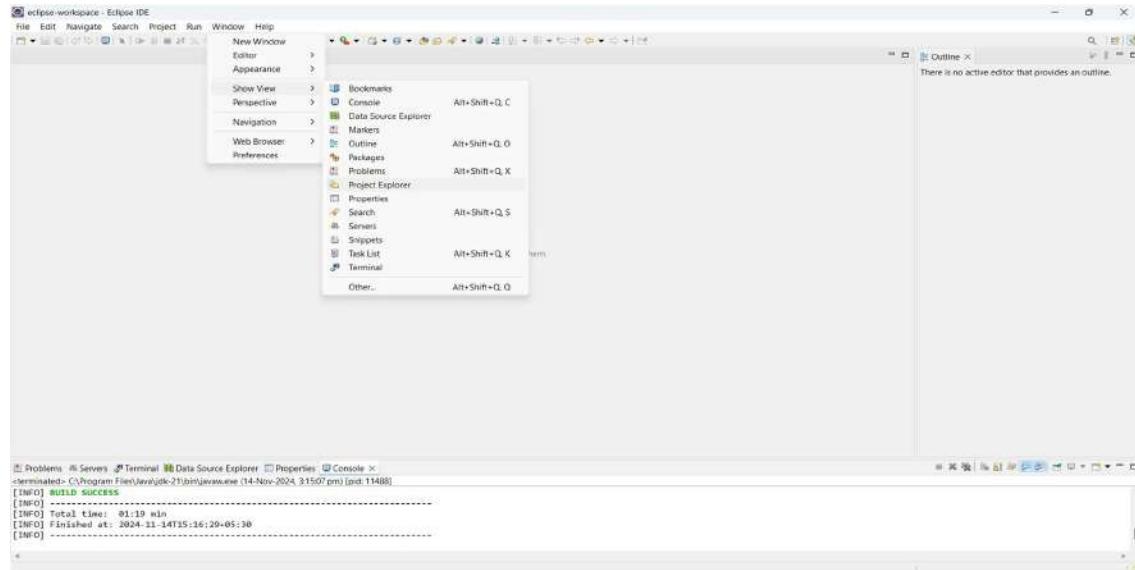


The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - Eclipse IDE". The central area displays the Maven CLI output in the "Console" tab, identical to the previous screenshot:

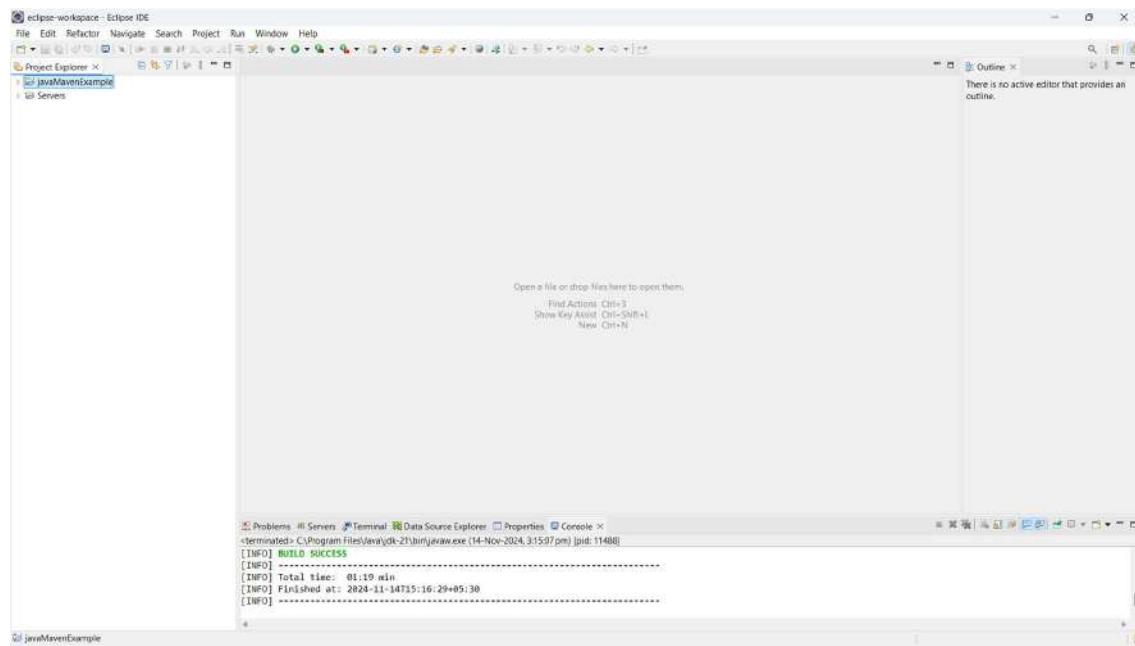
```
[INFO] [INFO] -- archetype: 0.3.1:generate {default-cli} @ standalone-pom ***
[INFO] [INFO] Generating project in interactive mode
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml
Progress (1): 0/16 MB
Progress (1): 0/16 MB
y
```

The status bar at the bottom right indicates "Creating maven-archetype-quickstart (33%)".

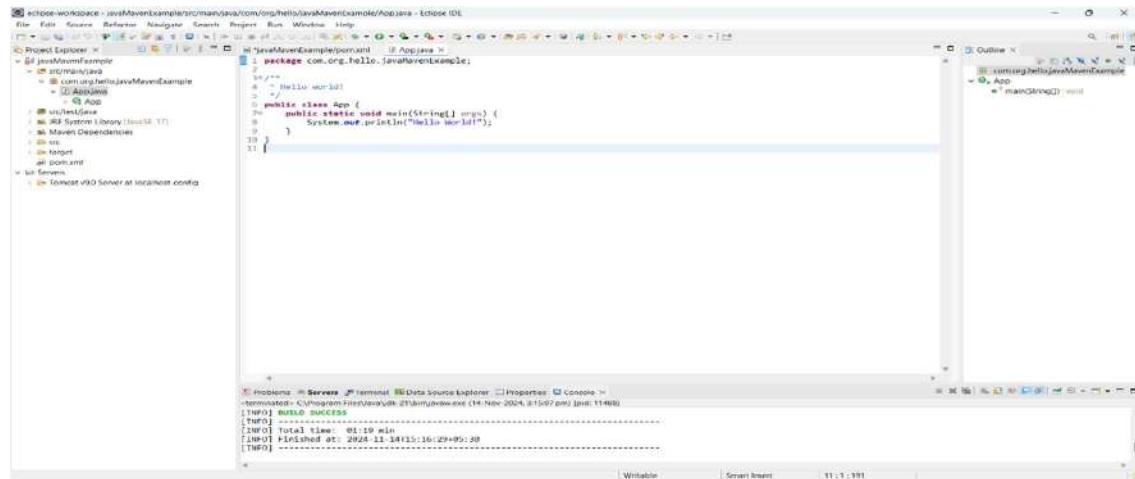
Open project explorer following the below steps



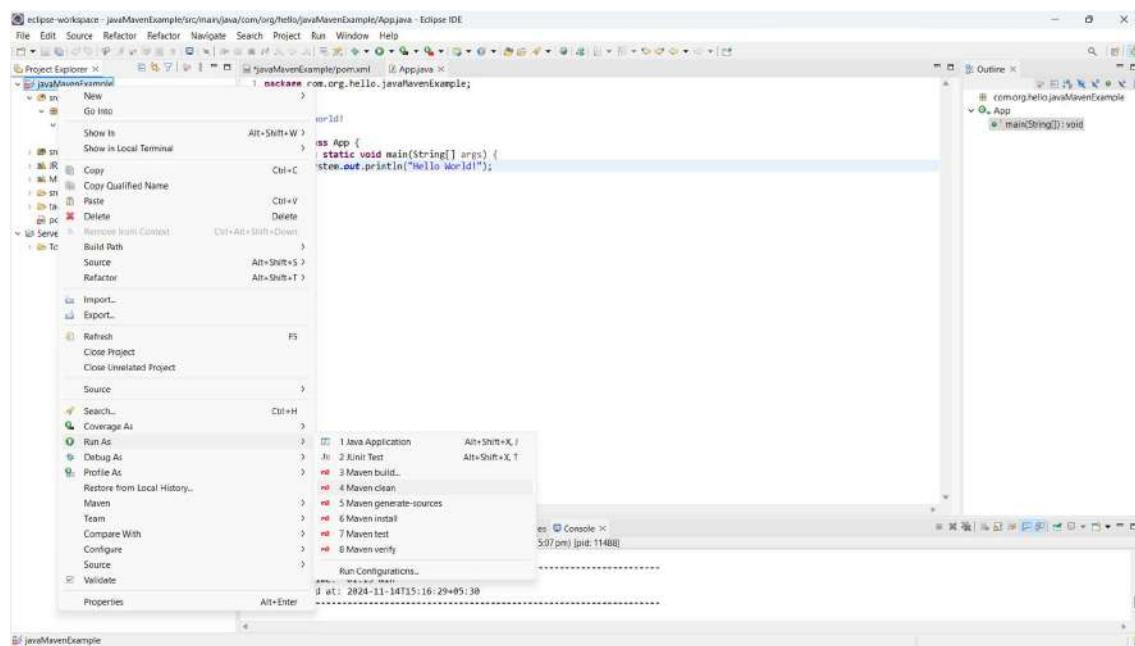
Click on the arrow along with your project for a dropdown



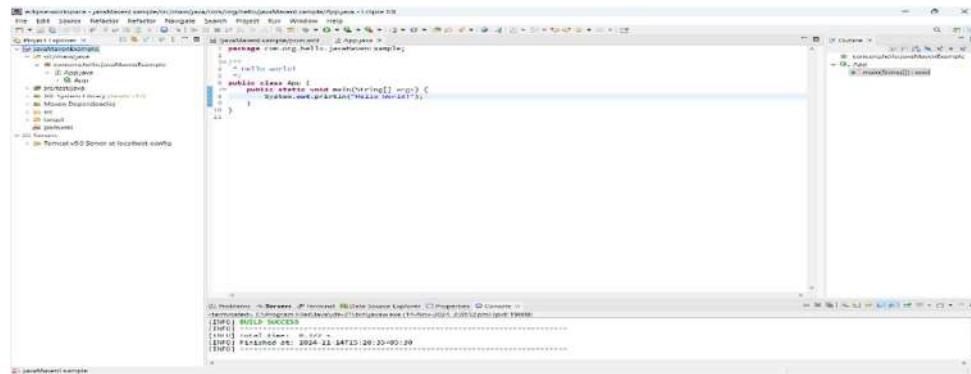
In the created project, Check for App.java file containing Hello World program



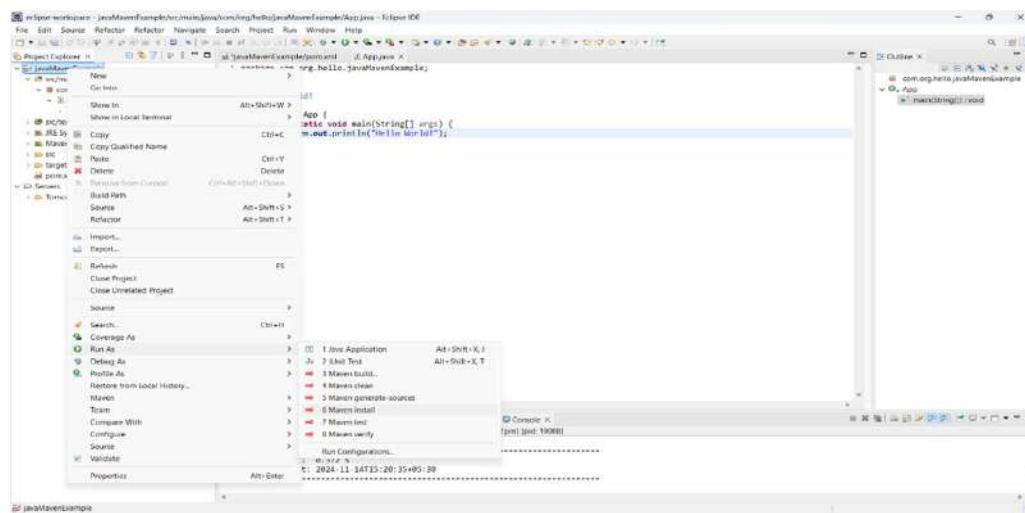
Right click on your project and Run As Maven Clean - It clears out the existing classes that you compiled from last compile



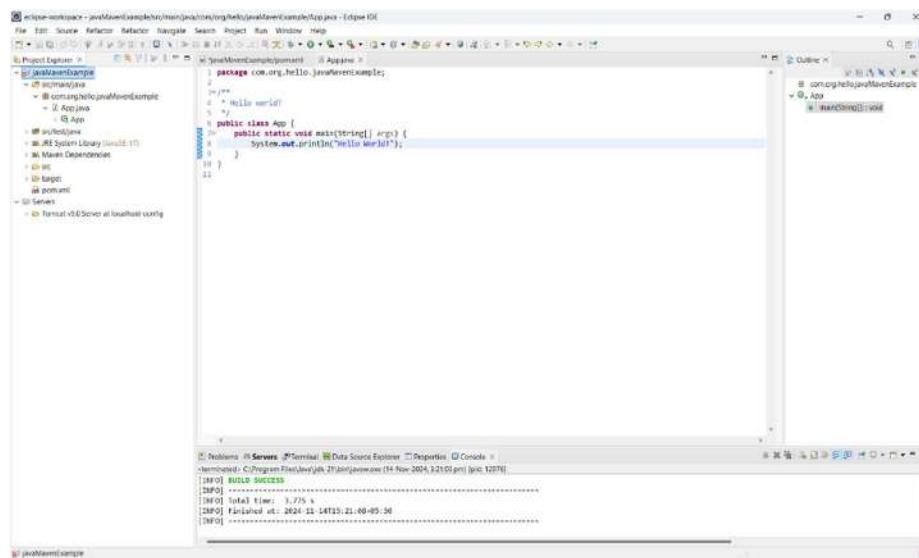
Check the console for Build Success



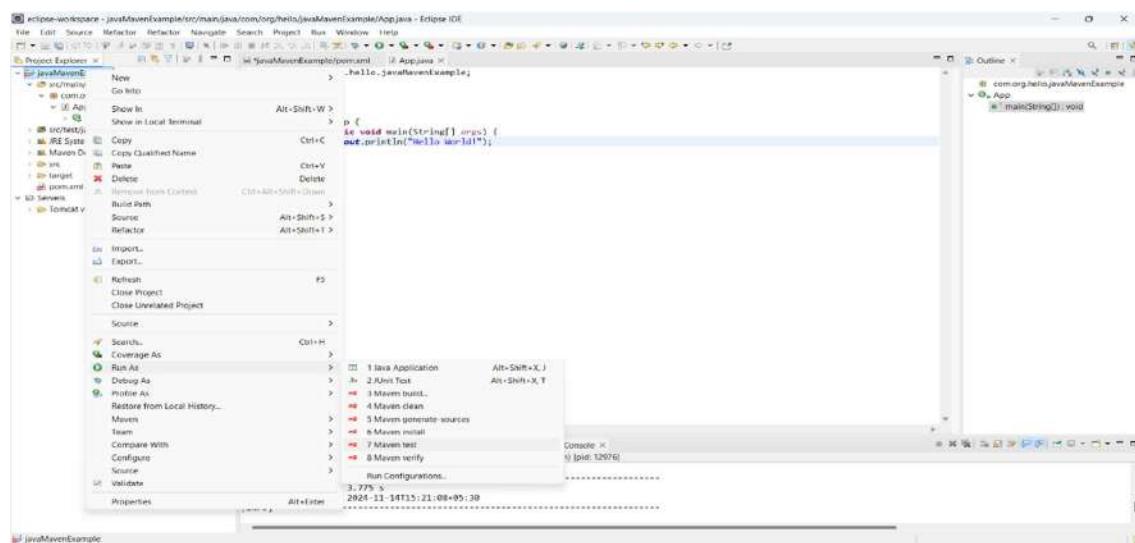
Right click on your project and Run As Maven Install - to add artifact(s) to the local repository



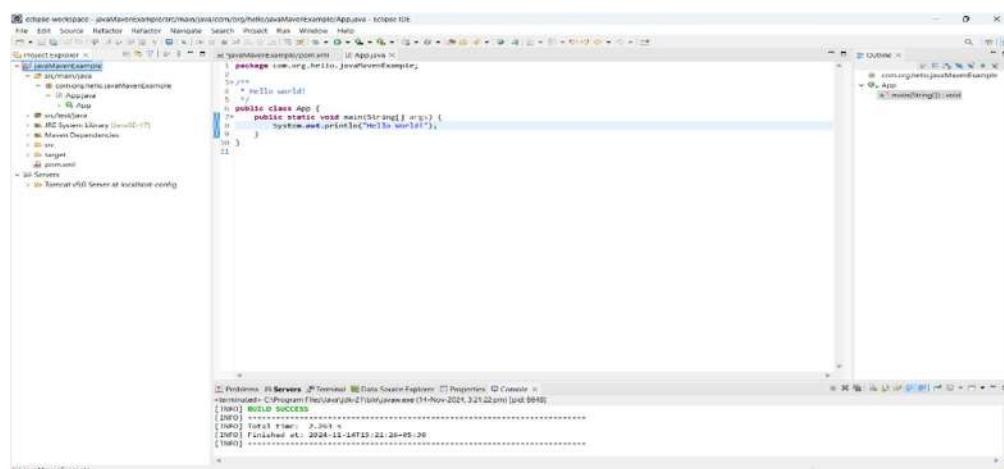
Check the console for Build Success



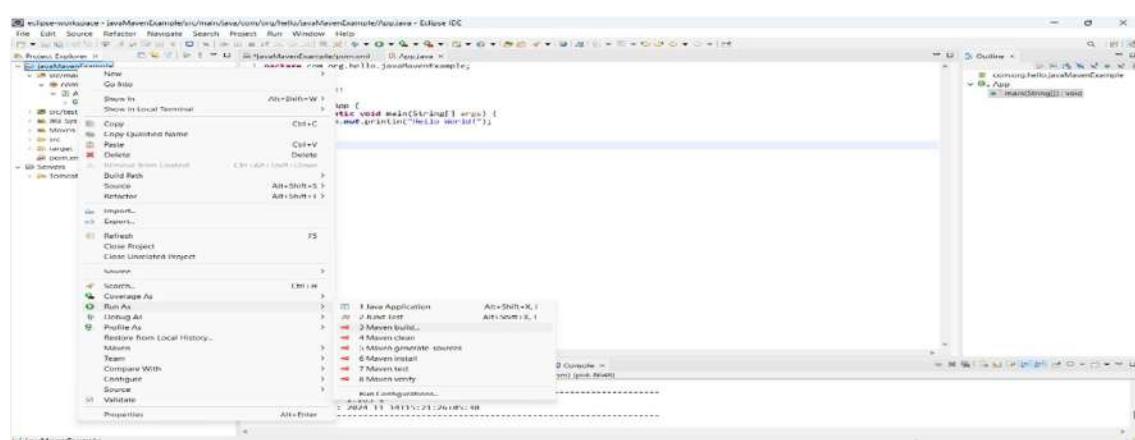
Right click on your project and Run As Maven test



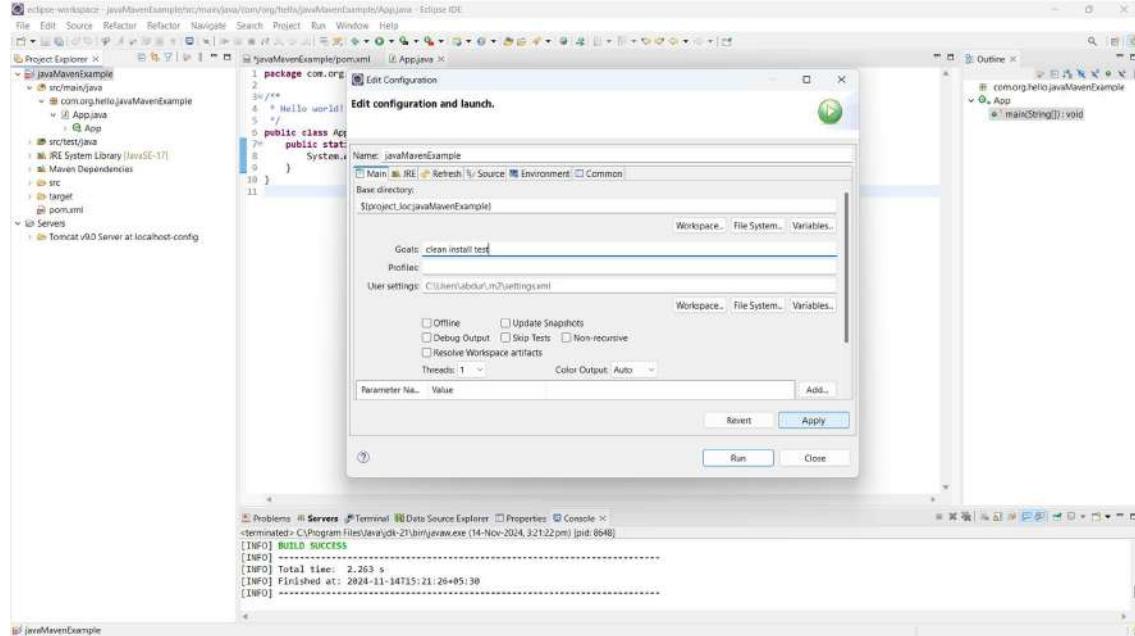
Check the console for Build Success



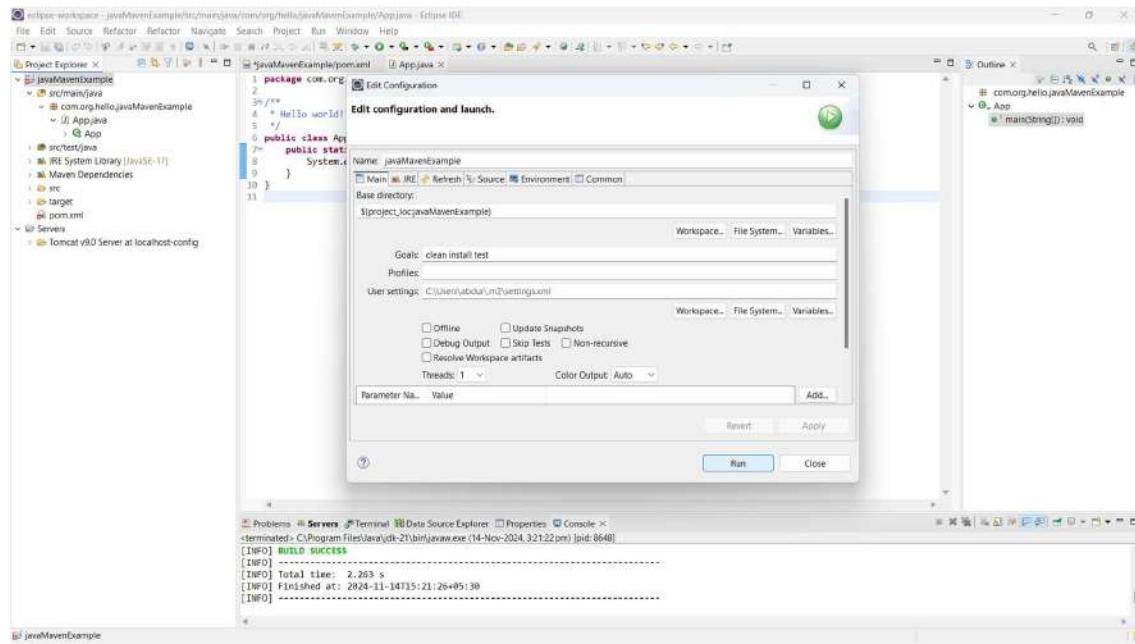
Right click on your project and Run As Maven build



Specify the goals as – Clean Install Test -> Click on Apply
-> Apply



Click on Run



Check the console for Build success

The screenshot shows the Eclipse IDE interface. The top window displays the Java code for `App.java`:

```

1 package com.org.hello.javaMavenExample;
2
3 /**
4 * Hello world!
5 */
6 public class App {
7     public static void main(String[] args) {
8         System.out.println("Hello World!");
9     }
10}

```

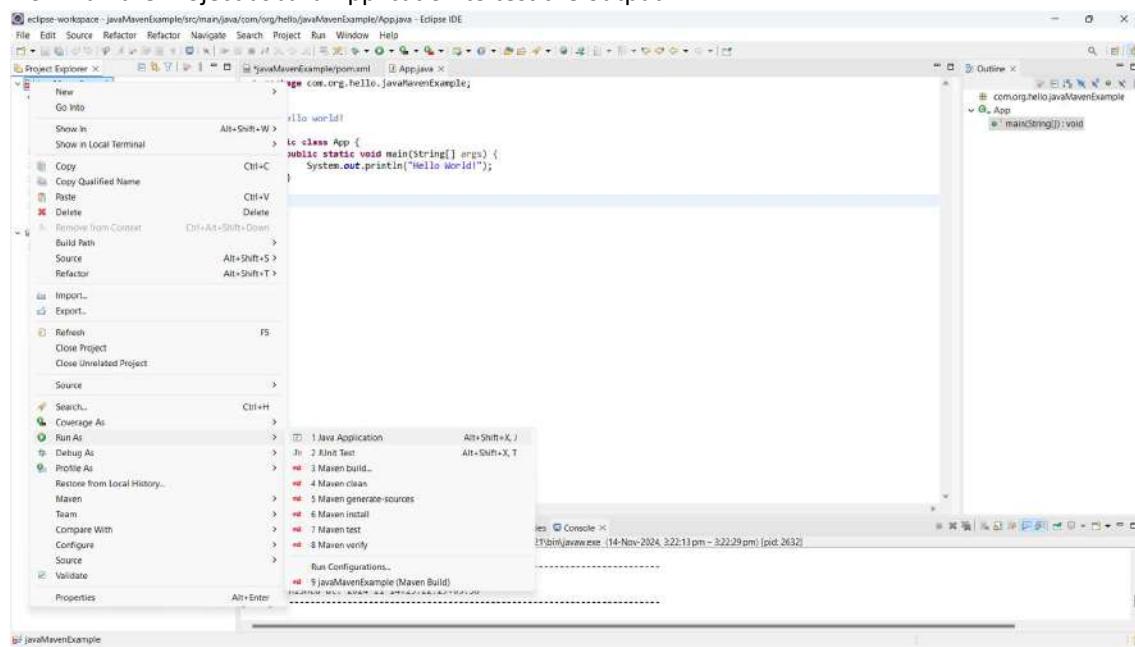
The bottom window shows the `Console` view with the following build log:

```

[INFO] BUILD SUCCESS
[INFO] Total time: 3.884 s
[INFO] Finished at: 2024-11-14T15:22:29+05:30
[INFO]

```

Now Run the Project as Java Application to test the output



In the console we can view the output of the project

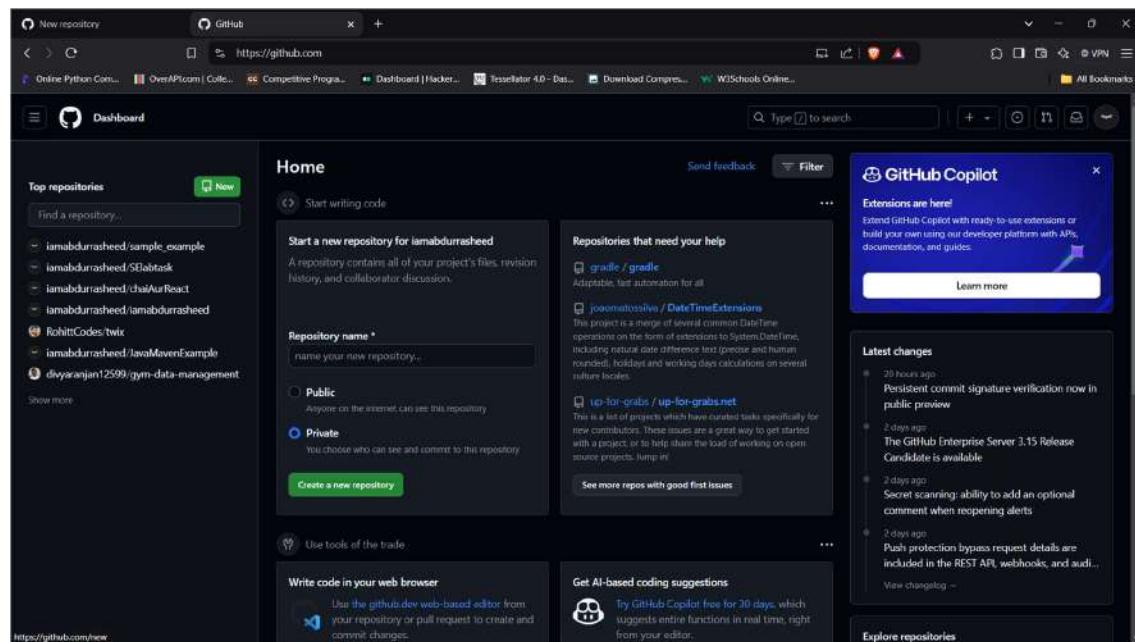
```

eclipse-workspace - javaMavenExample/src/main/java/com/org/hello/javaMavenExample/App.java - Eclipse IDE
File Edit Source Refactor Refactor Navigate Project Run Window Help
Project Explorer X Servers X Data Source Explorer X Properties X Console X
javaMavenExample/pom.xml App.java
1 package com.org.hello.javaMavenExample;
2
3 /**
4 * Hello world!
5 */
6 public class App {
7     public static void main(String[] args) {
8         System.out.println("Hello World!");
9     }
10 }
11

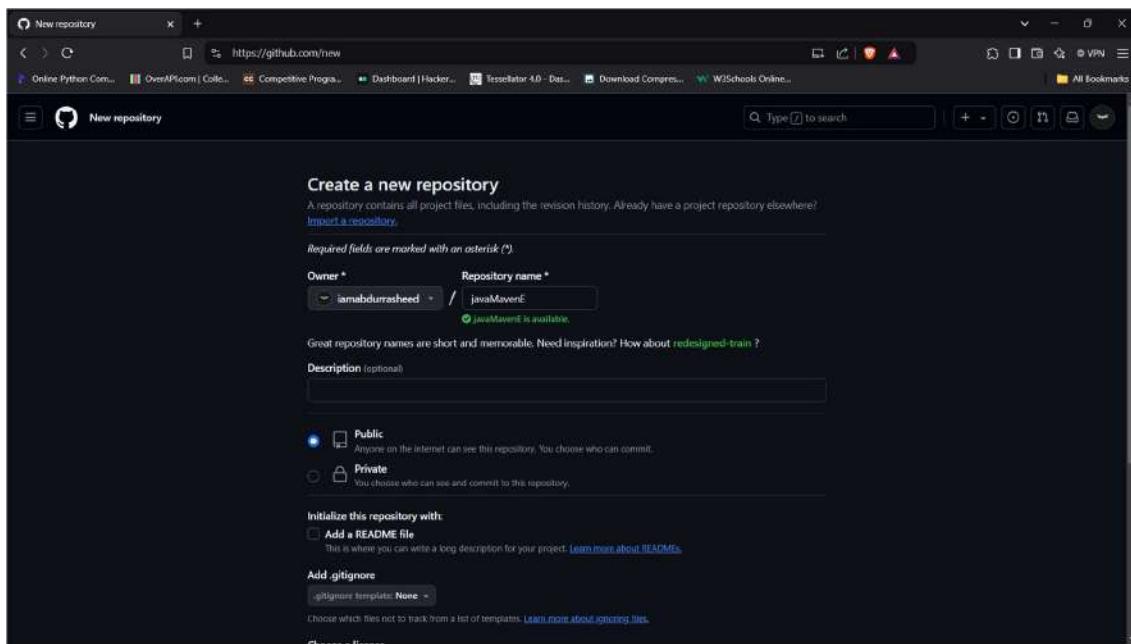
```

Problems Servers Data Source Explorer Properties Console
App [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (14-Nov-2024, 3:26:28 pm - 3:26:28 pm) [pid: 22296]
Hello World!

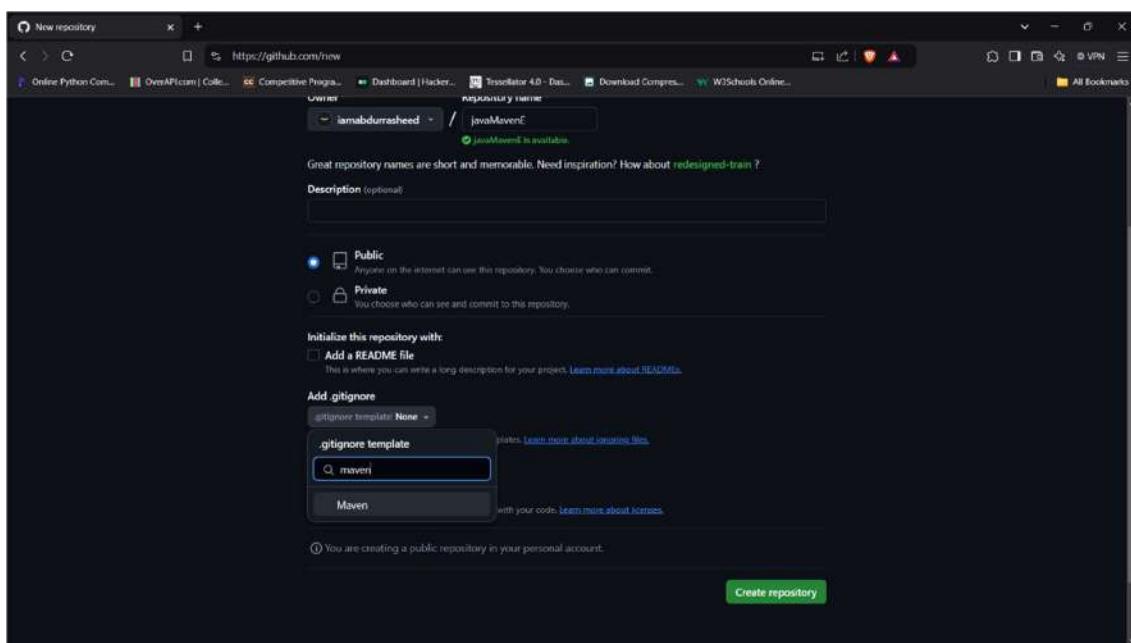
Next is to push the project into GitHub, for this, create a new repository in GitHub



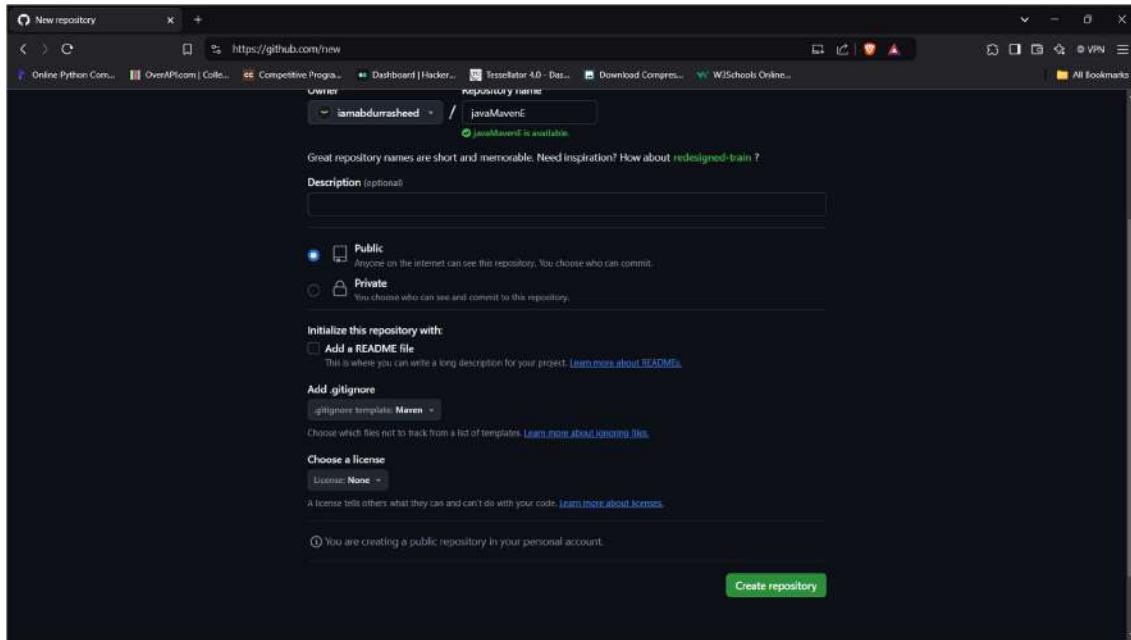
Enter the required fields



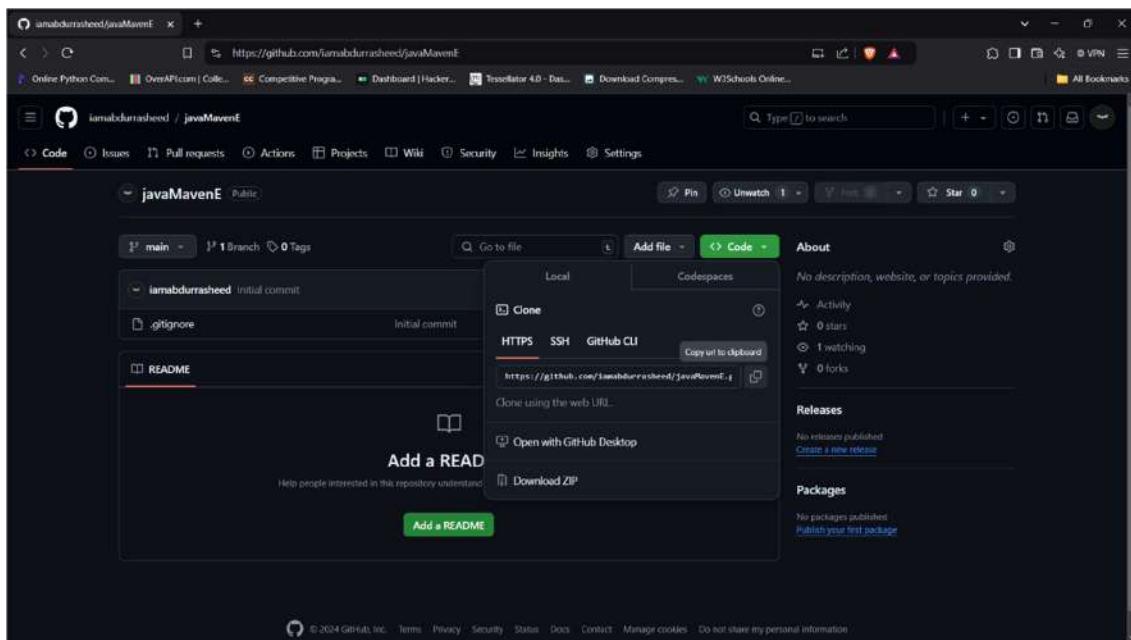
Select Maven in gitignore- to ignore unwanted target files in maven project while committing



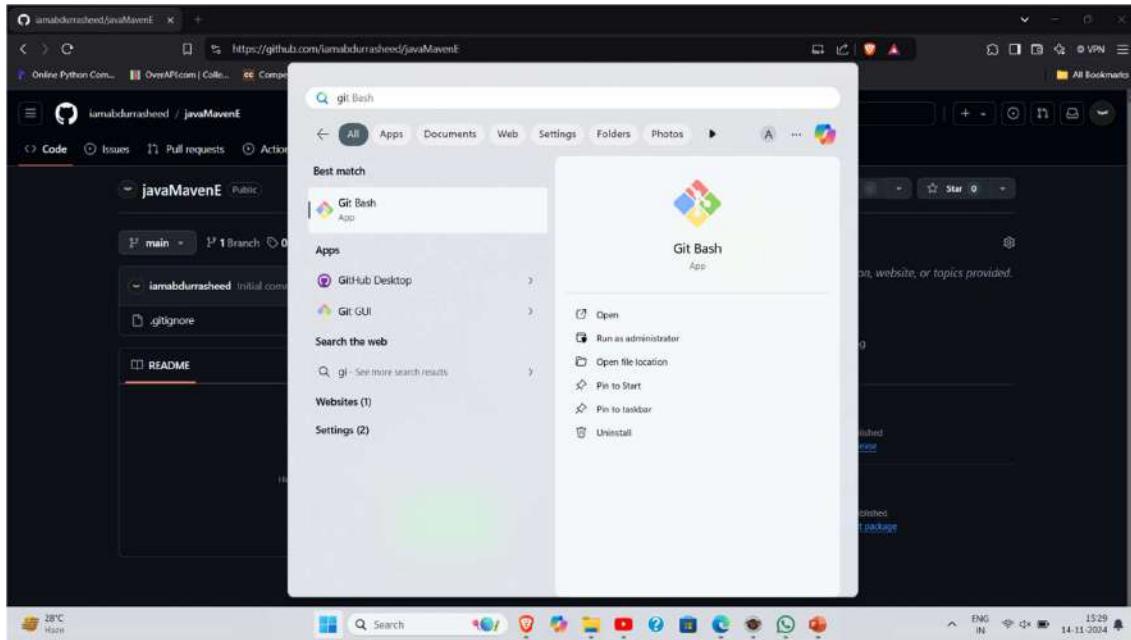
Click on Create repository



Copy the HTTPS URL



Open Git Bash to push the project into GitHub



Clone the GitHub repository into the local system

```

TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~
$ git clone https://github.com/iamabdurrashed/javaMavenE.git
Cloning into 'javaMavenE'...
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 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

```

The terminal window shows the command \$ git clone https://github.com/iamabdurrashed/javaMavenE.git being run, followed by the output of the cloning process. The directory structure of the cloned repository is also visible on the right side of the terminal window.

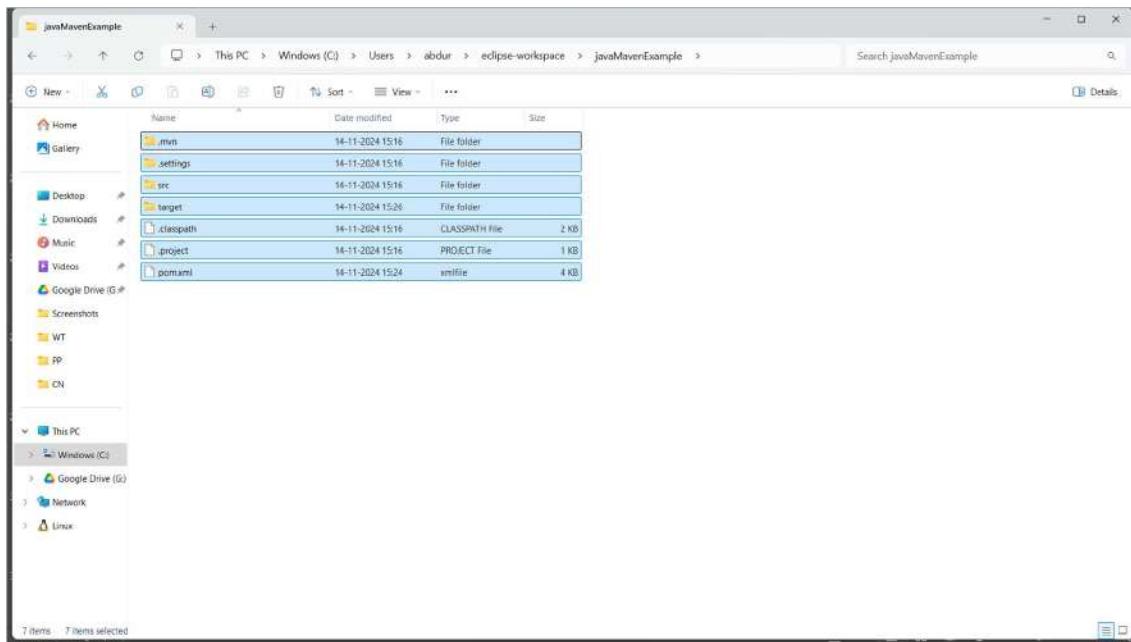
After cloning follow the steps as shown

```

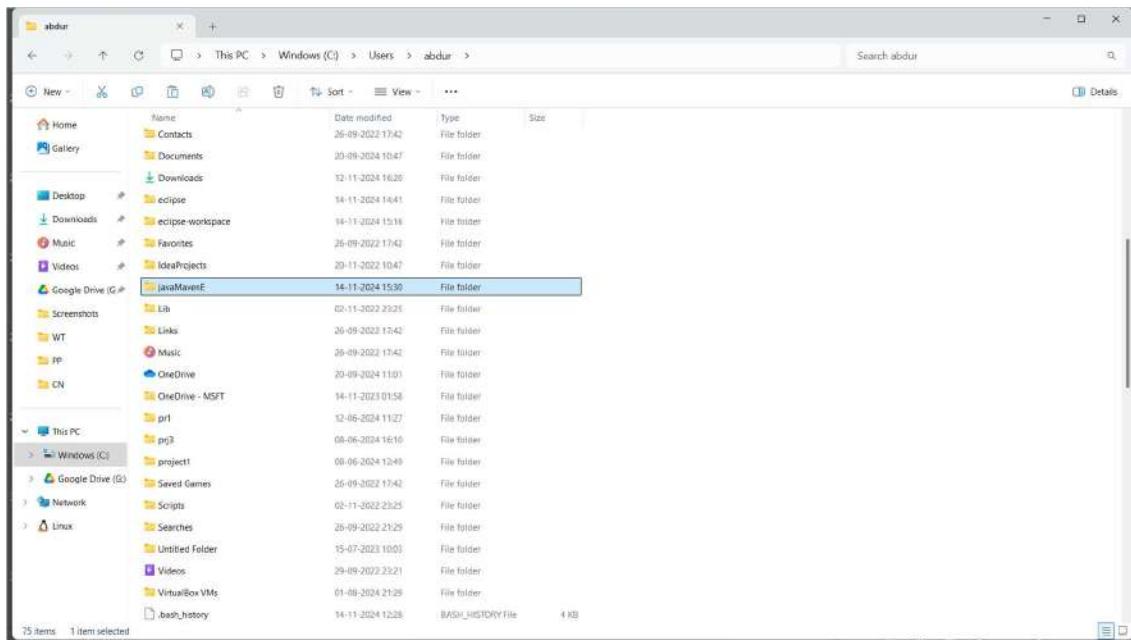
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~
$ cd javaMavenE
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/javaMavenE (main)
$ |

```

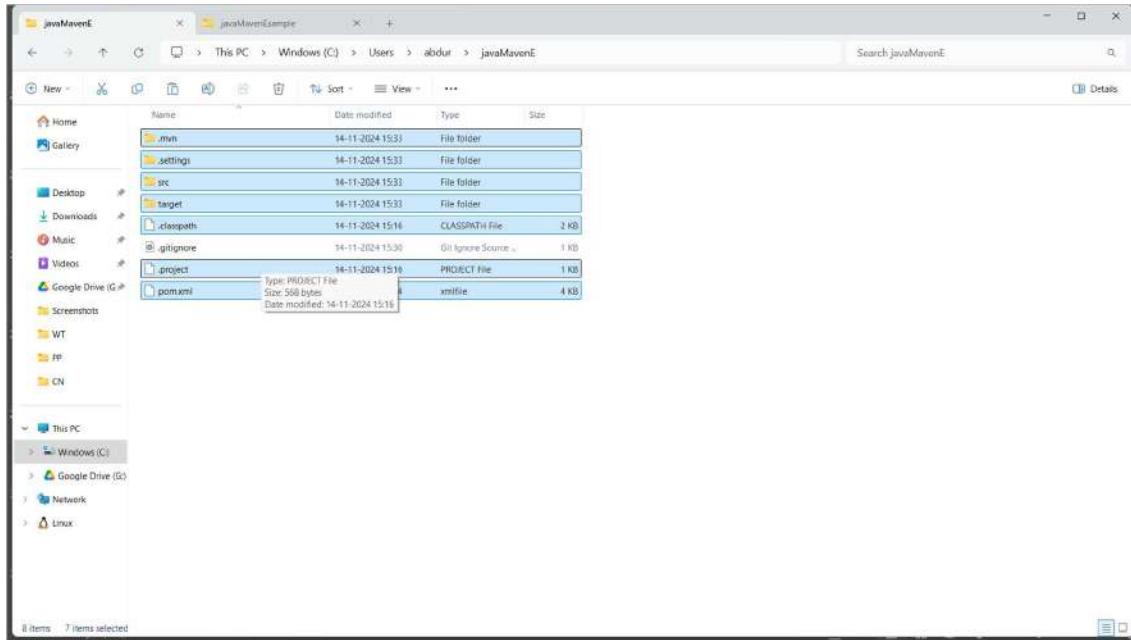
One way of cloning the local and global projects is to copy the contents from eclipse workspace to the local cloned repository as shown



Scroll down in users → Username to find the cloned repo



Paste the copied files into the cloned repo



Using Git bash, add and commit the changes made as shown
then using git push push it into remote repository

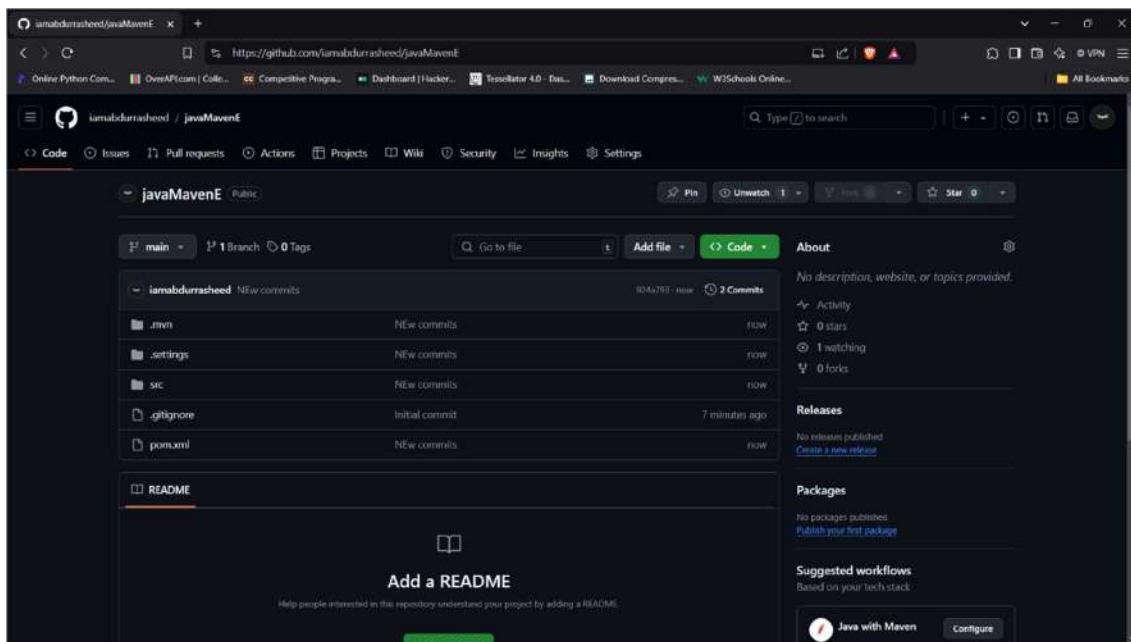
```

MINGW64:/c/Users/abdur/javaMavenE
$ git add .
TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/javaMavenE (main)
$ git commit -m "New commits"
[main 924a793] New commits
 8 files changed, 155 insertions(+)
 create mode 100644 .mvn/jvm.config
 create mode 100644 .mvn/maven.config
 create mode 100644 .settings/org.eclipse.core.resources_prefs
 create mode 100644 .settings/org.eclipse.jdt.core_prefs
 create mode 100644 .settings/org.eclipse.m2e.core_prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/java/com/org/hello/javaMavenExample/App.java
 create mode 100644 src/test/java/com/org/hello/javaMavenExample/AppTest.java

TAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/javaMavenE (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) 62 KiB | 537.00 KiB/s, done.
Total 24 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/tamabdurrasheed/javaMavenE.git
 b4ec3bb..924a793 main..> main

```

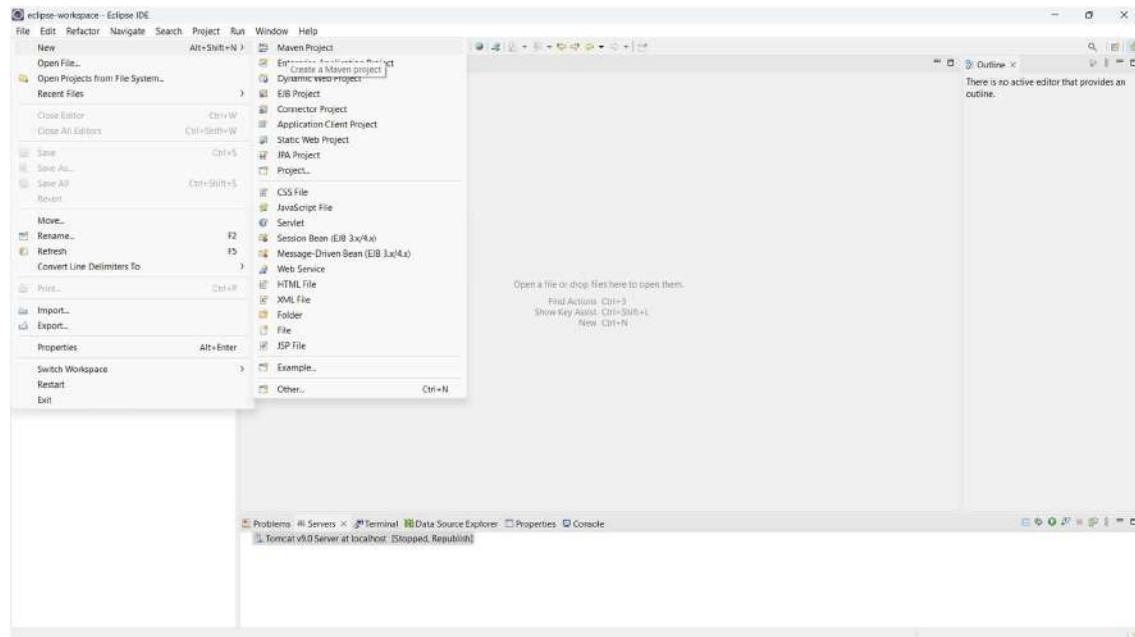
Refresh the GitHub to see the updated repository



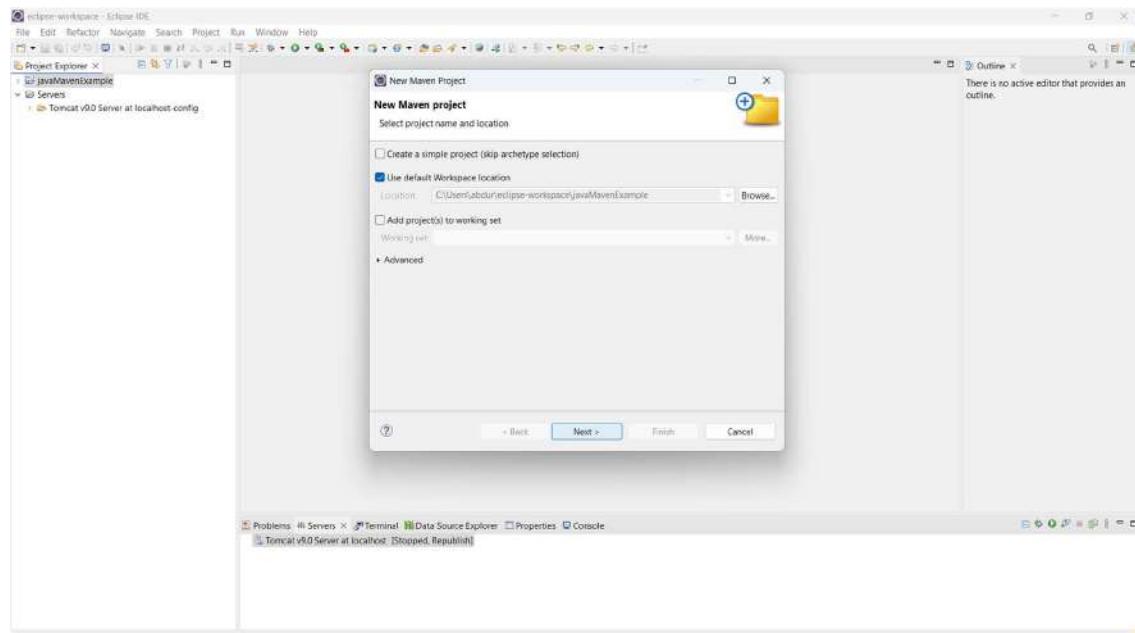
Repository is successfully updated

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

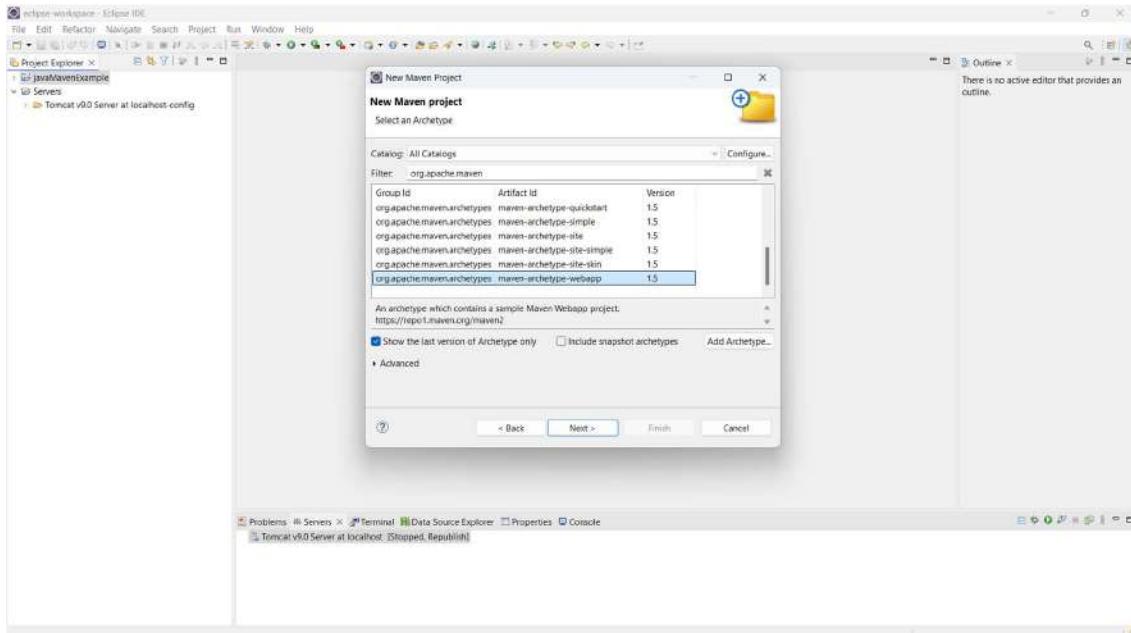
Create a Maven Project - > File -> New -> Maven Project



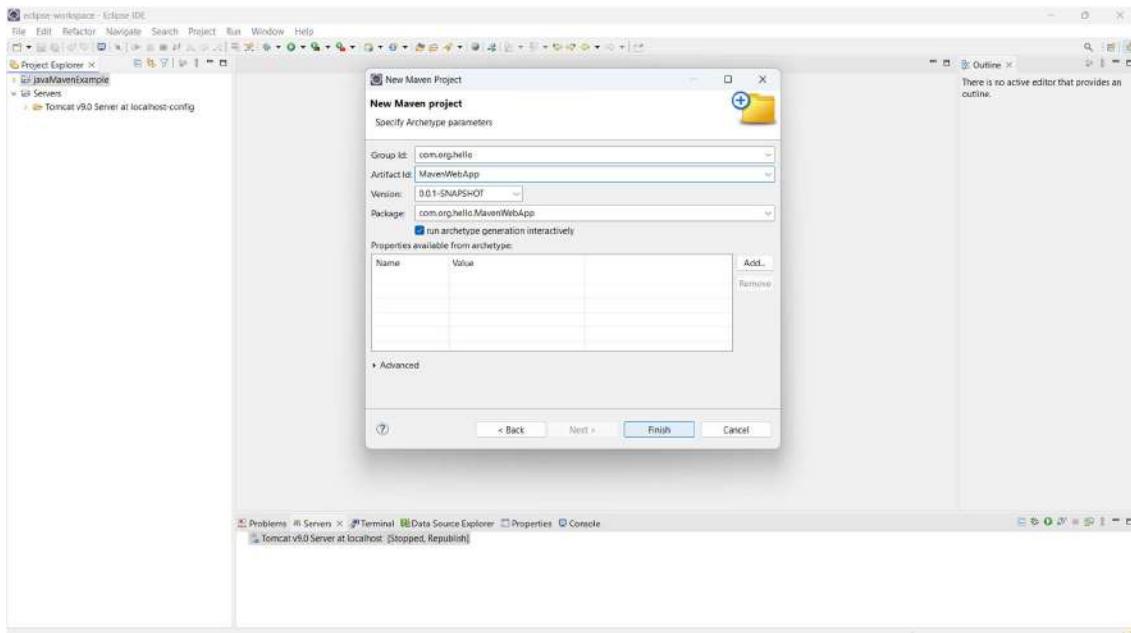
Click on Next



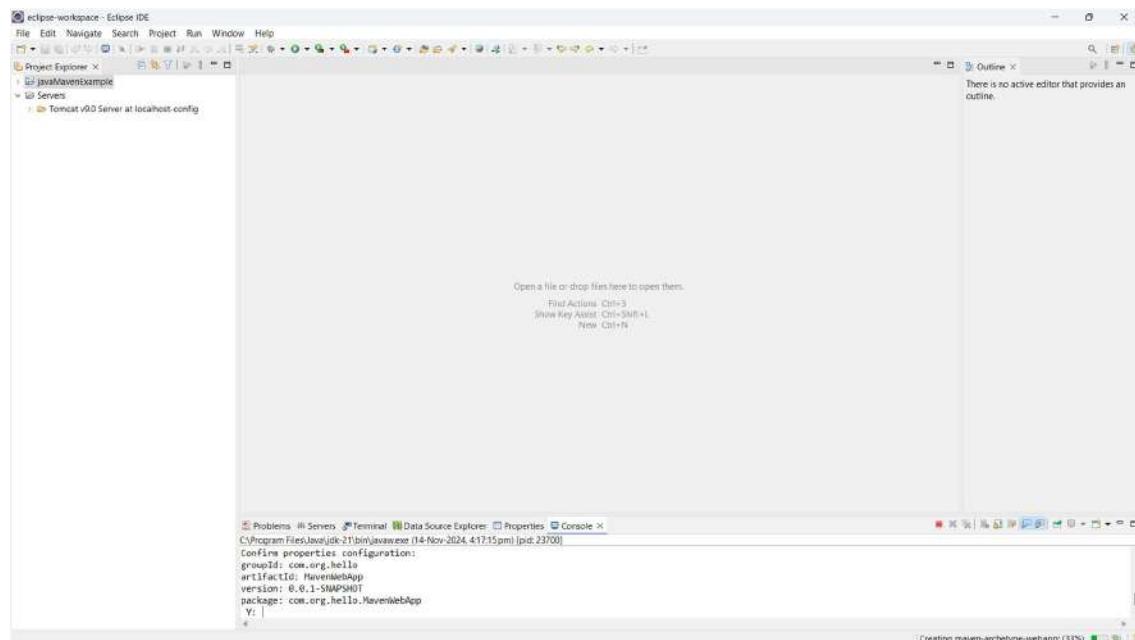
Select the webapp archetype under org.apache.maven.archetypes and click on Next



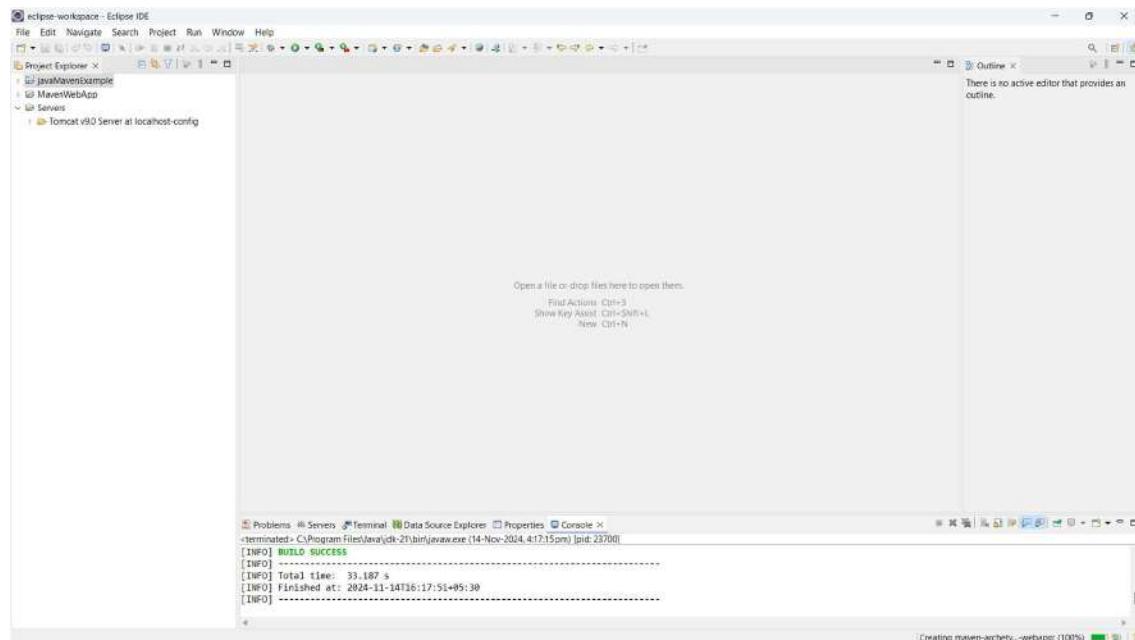
Enter some GroupId and Artifact Id → Finish



Click beside y: and tap on enter



Project successfully created



Open browser and search for servlet api maven dependency
click on the first link

Showing results for servlet api maven *dependency*
Search instead for servlet api maven dependency

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

javax.servlet > servlet-api - Maven Repository
Java Servlet is the foundation web specification in the Java Enterprise Platform. Developers can build web applications using the Servlet API to interact ...

- Java Servlet API » 4.0.1**
Java Servlet API » 4.0.1. Java Servlet is the foundation web ...
- 3.1.0**
Java Servlet API » 3.1.0. Java Servlet is the foundation web ...
- 2.5**
JavaServer(TM) Specification » 2.5. Java Servlet is the foundation ...
- Java Servlet**
Java Servlet is the foundation web specification in the Java ...
- 3.0-alpha-1**
JavaServer(TM) Specification » 3.0-alpha-1. Java Servlet is the ...

Scroll for the versions..

MVN REPOSITORY

Indexed Artifacts (45.6M)

JavaServlet(TM) Specification

Java Servlet is the foundation web specification in the Java Enterprise Platform. Developers can build web applications using the Servlet API to interact with the request/response workflow.

License: CDDL, GPL, GPL 2.0

Categories: Java Specifications

Tags: standard, servlet, javax, api, specs

Ranking: #49 in MvnRepository (See Top Artifacts)
#2 in Java Specifications

Used By: 13,399 artifacts

This artifact was moved to:
[javax.servlet + javax.servlet-api](#)

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, 2008
2.4 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

Click on version 2.5because the latest versions are not always stable
to be on safer side we prefer the previous one

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

Scroll down after the page opens
and copy dependency

```
[<dependency>
<groupId>javax.servlet</groupId>
<artifactId>servlet-api</artifactId>
<version>3.0-alpha-1</version>
<scope>provided</scope>
</dependency>]
```

Copied to clipboard!

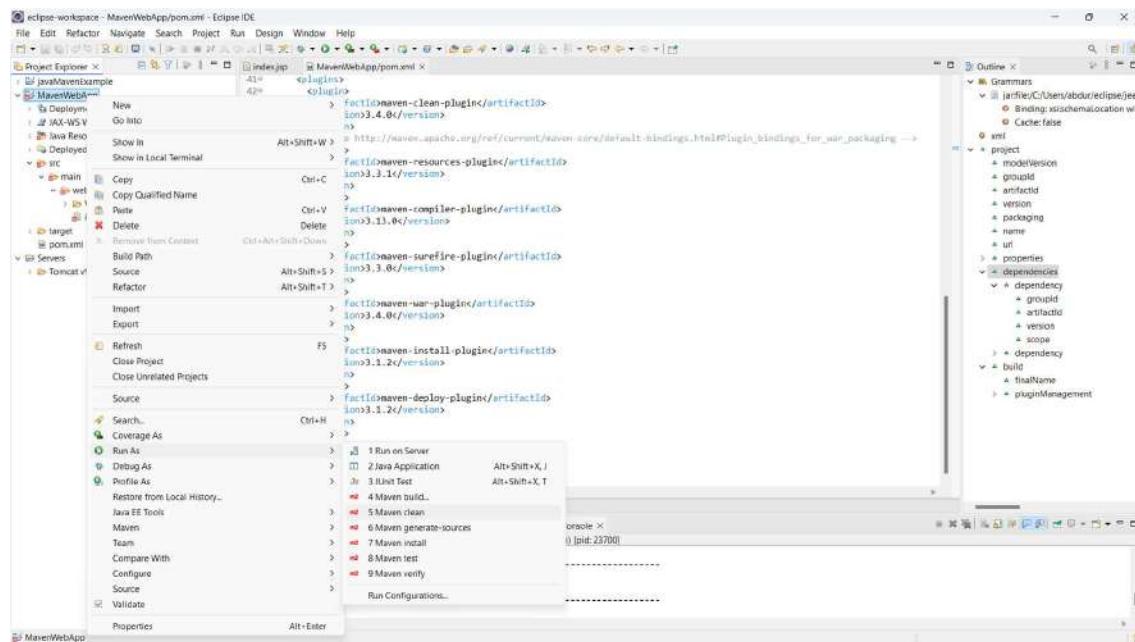
Now in the MavenWebApp → target → pom.xml
in <dependencies></ dependencies >
just before the closing tag paste it

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

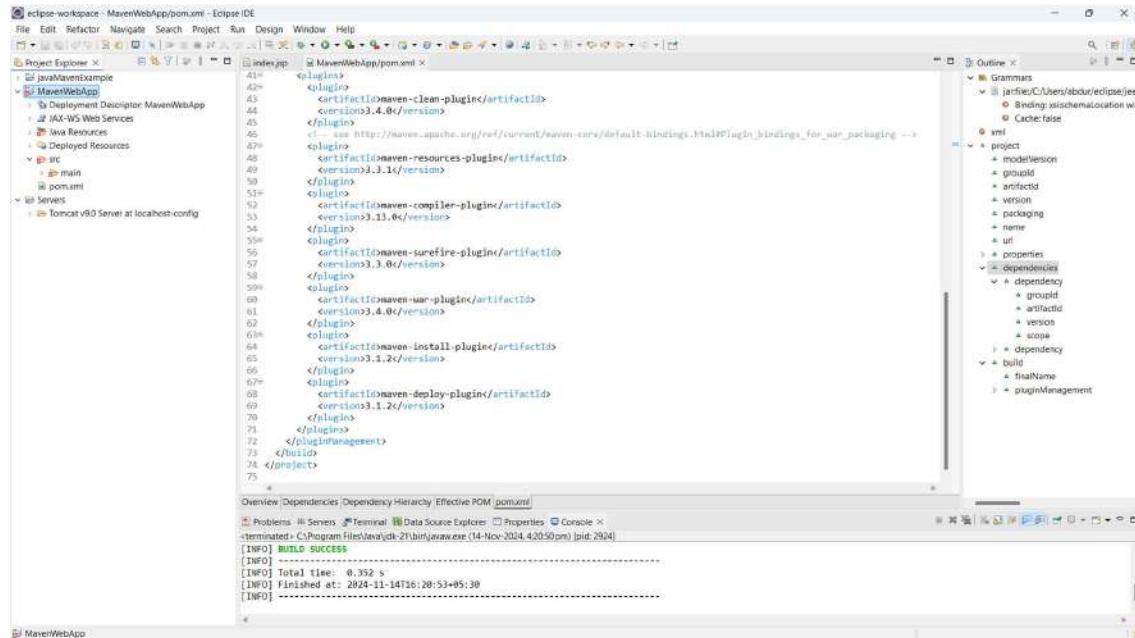
- Project Explorer View:** Shows the Maven project structure:
 - JavaMavenExample
 - MavenWebApp
 - Deployment Descriptor: MavenWebApp
 - JAX-WS Web Services
 - Java Resources
 - Deployed Resources
 - src
 - main
 - webapp
 - WEB-INF
 - index.jsp
 - target
 - pom.xml
- POM Editor View:** Displays the `pom.xml` file content.
- Outline View:** Shows the project structure and dependencies.
- Bottom Status Bar:** Shows the build status as `BUILD SUCCESS`.

Remove the comment and scope from the copied dependency

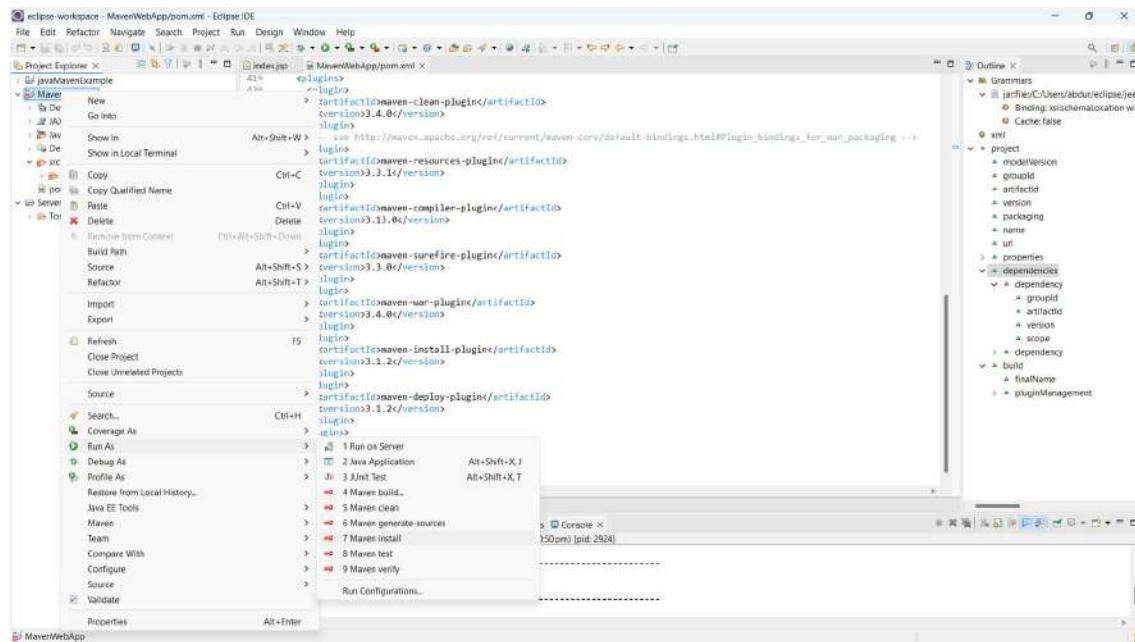
Right click on your project and Run As Maven Clean - It clears out the existing classes that you compiled from last compile



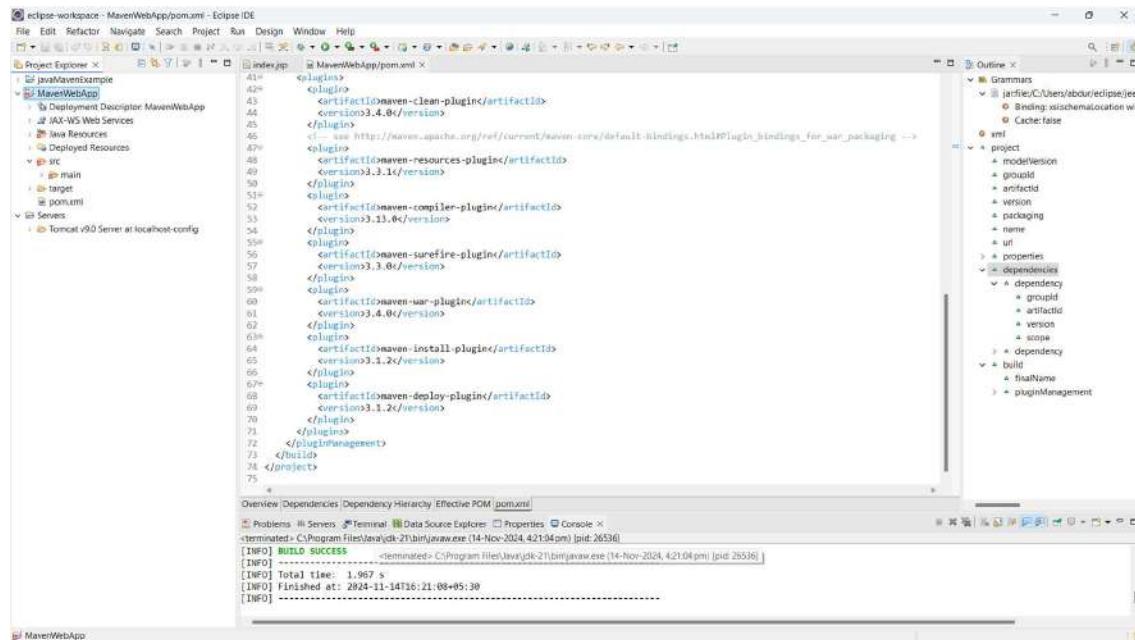
Check the console for Build success



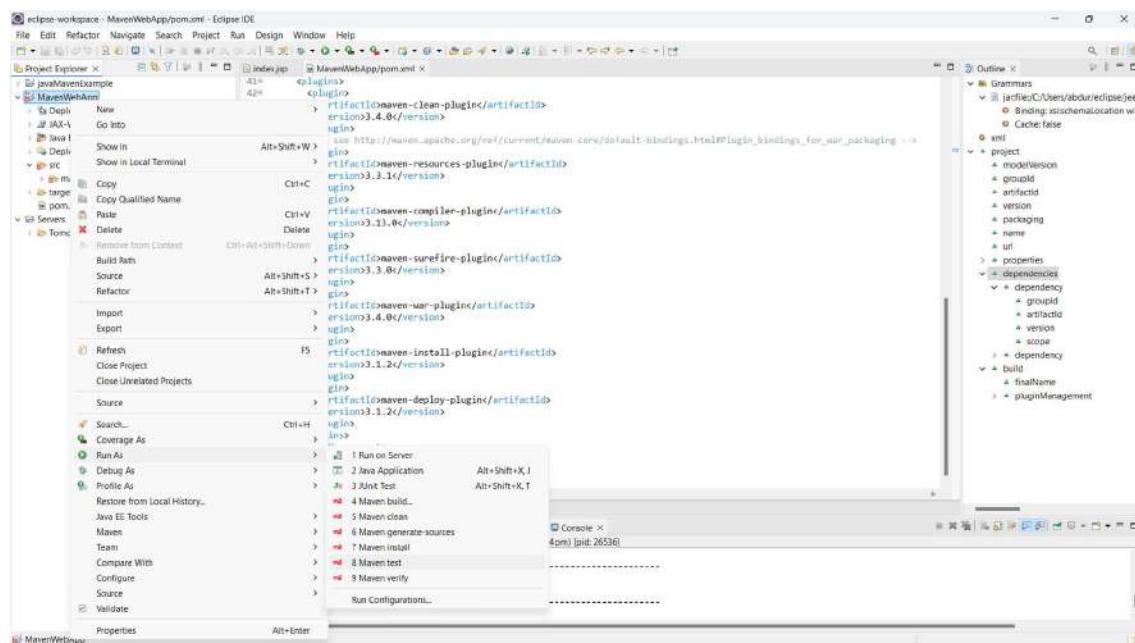
Right click on your project and Run As Maven Install - to add artifact(s) to the local repository



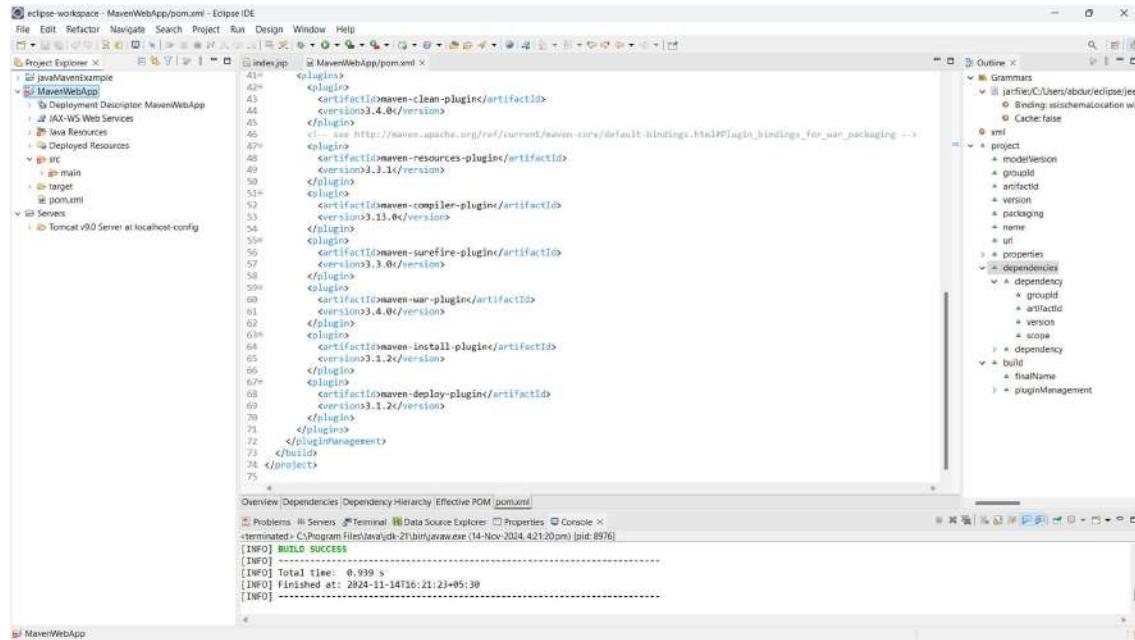
Check the console for Build success



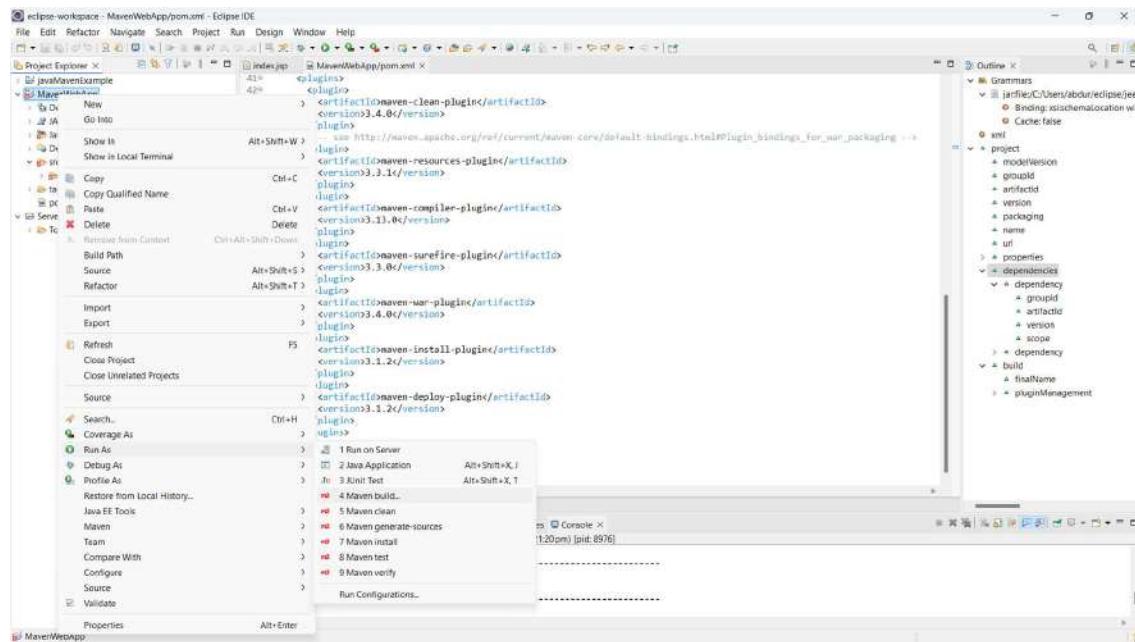
Right click on your project and Run As Maven test



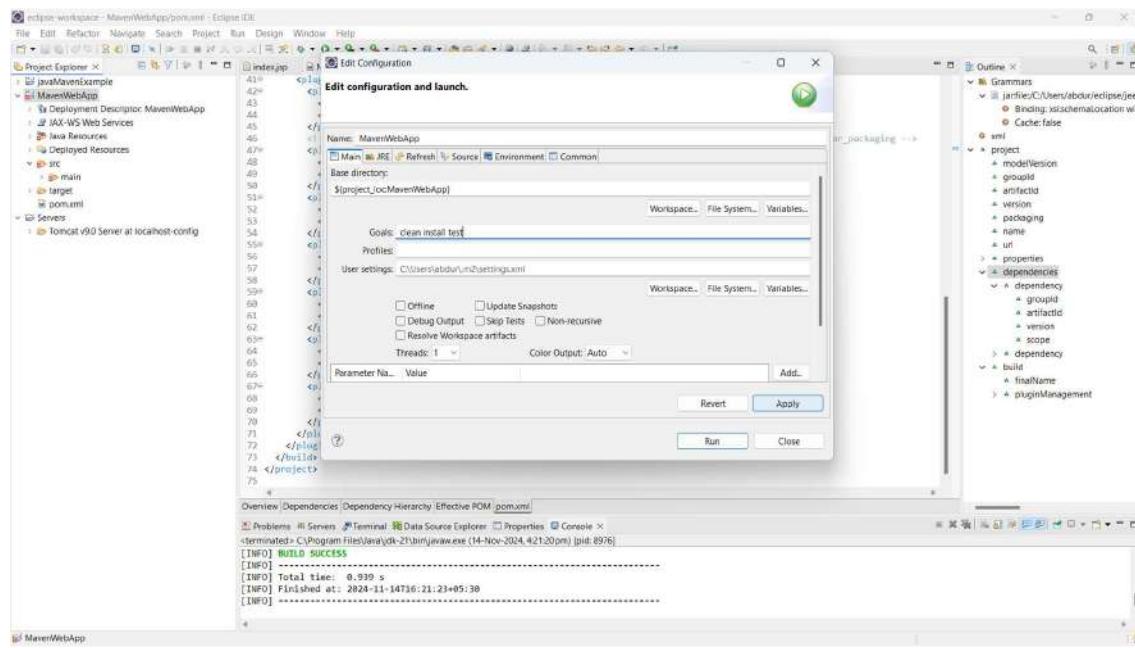
Check the console for Build success



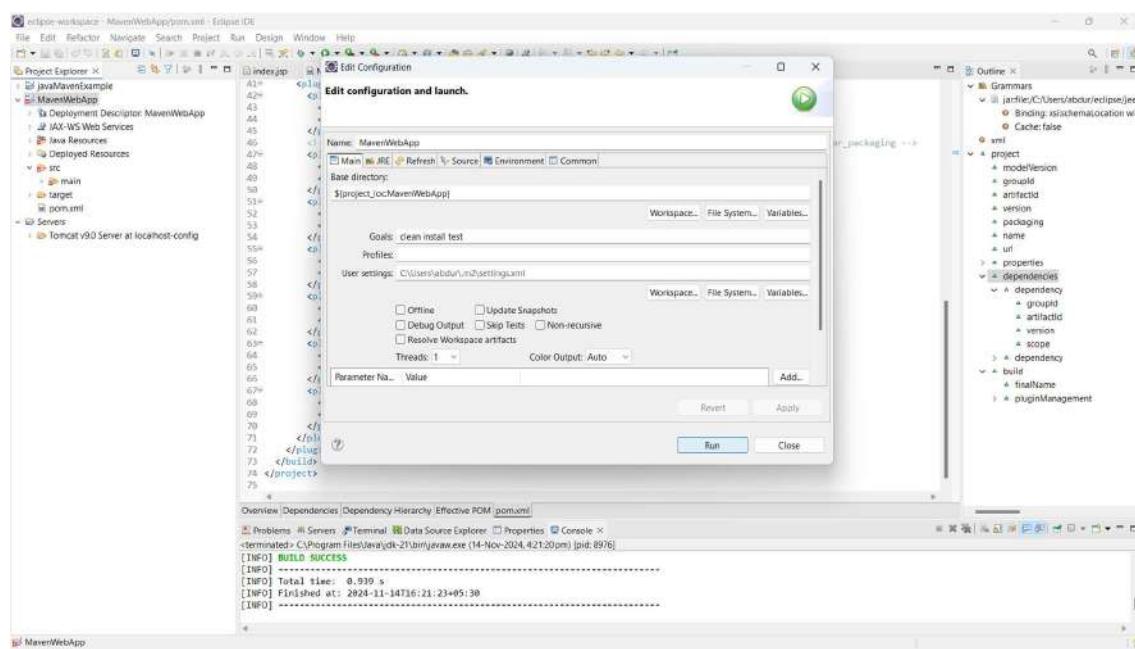
Right click on your project and Run As Maven build



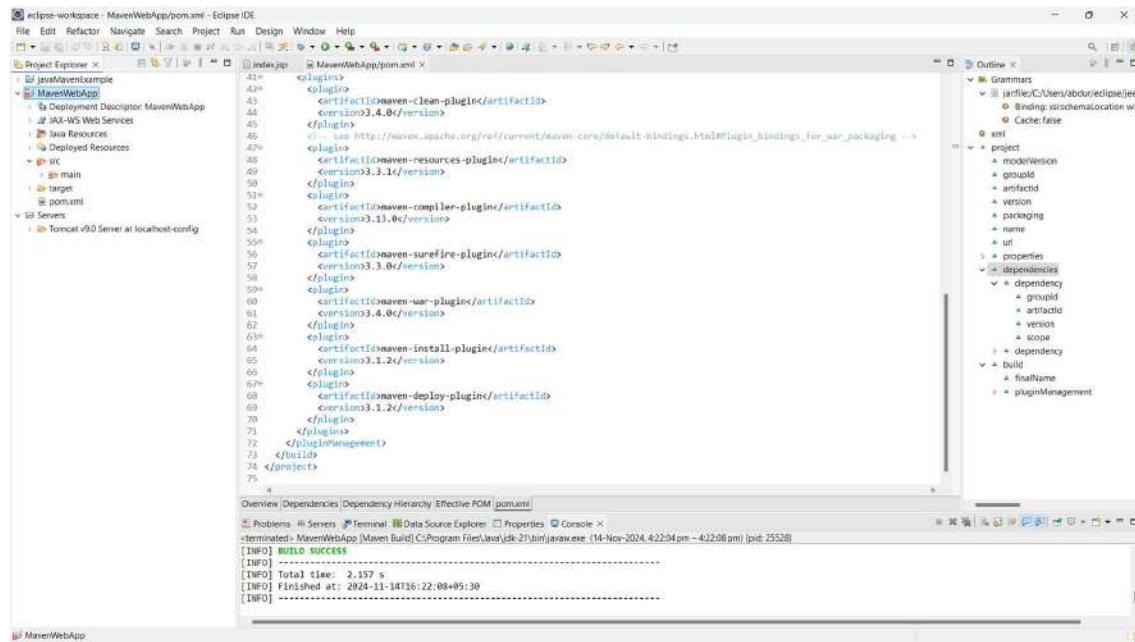
Specify the goals as – Clean Install Test -> Click on Apply
-> Apply



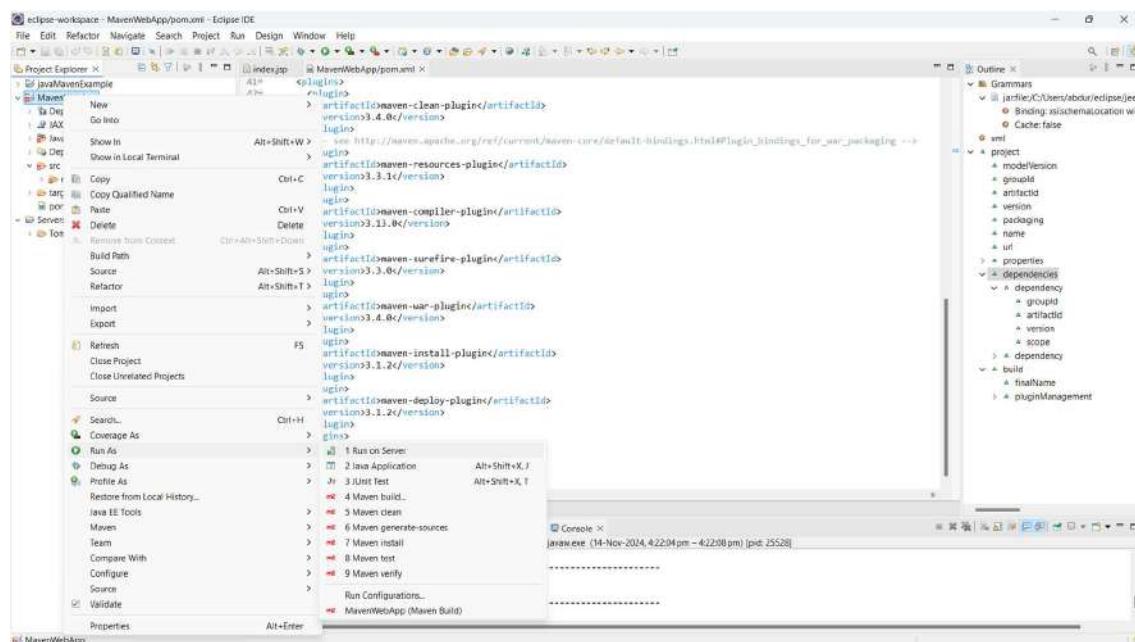
Click on Run



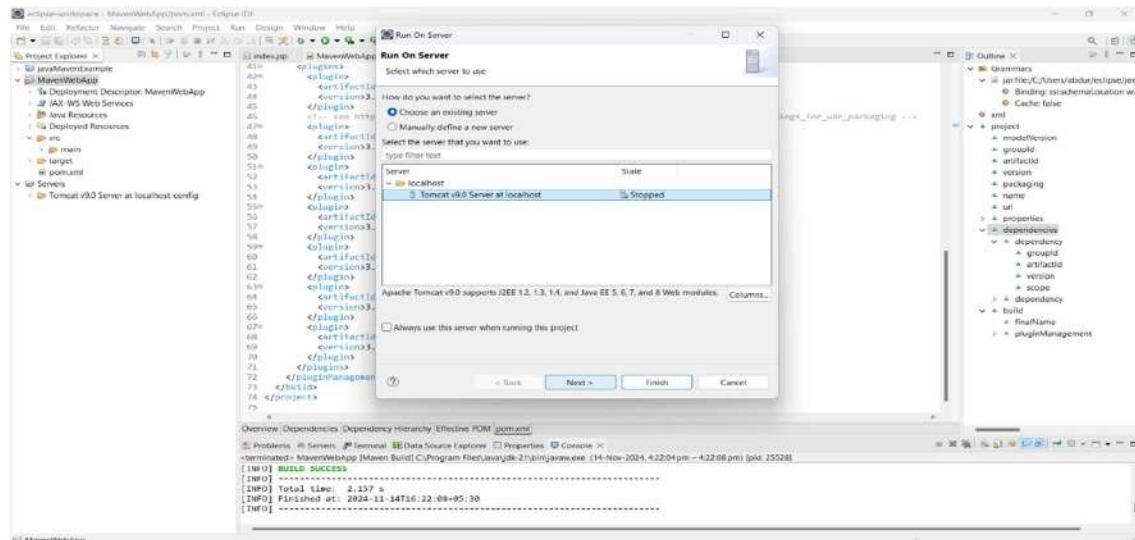
Check the console for Build success



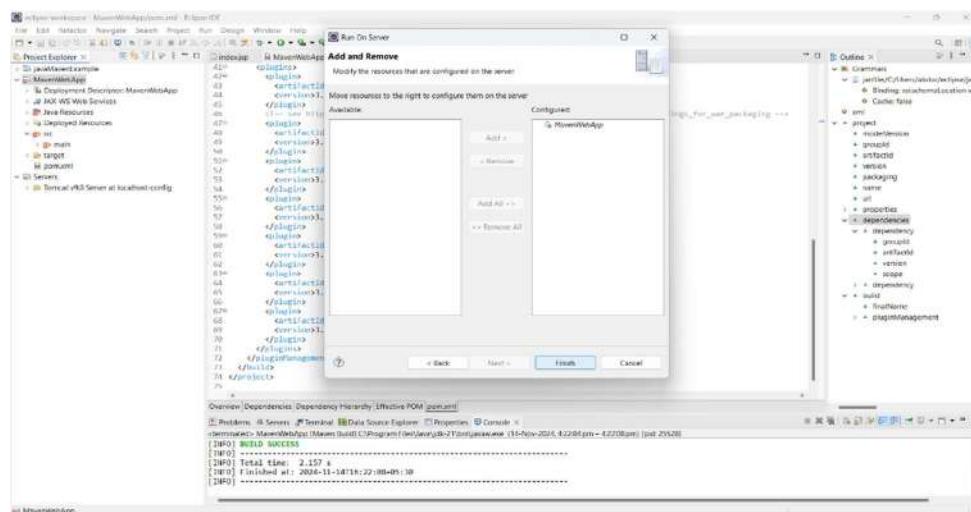
Right click on your project and Run As 1Run on server



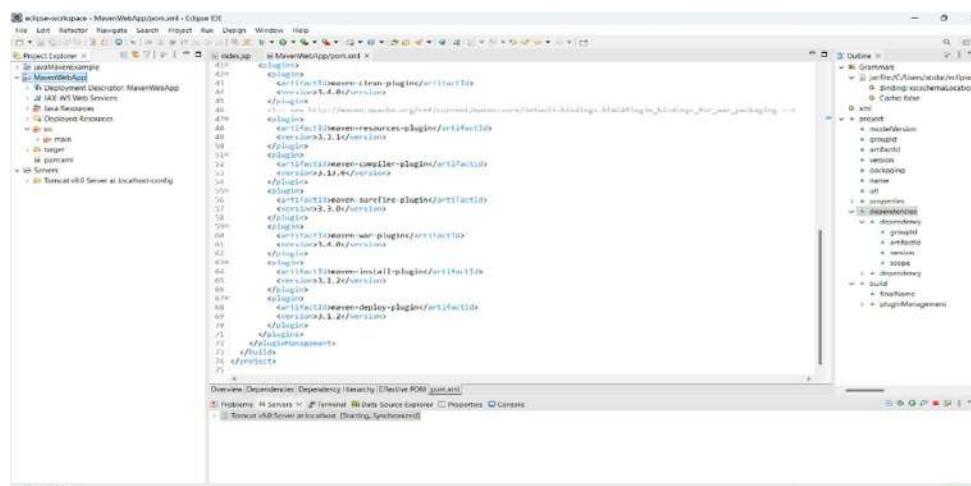
Click on installed tomcat v9.0 server → Next



Click on Finish



Tomcat started

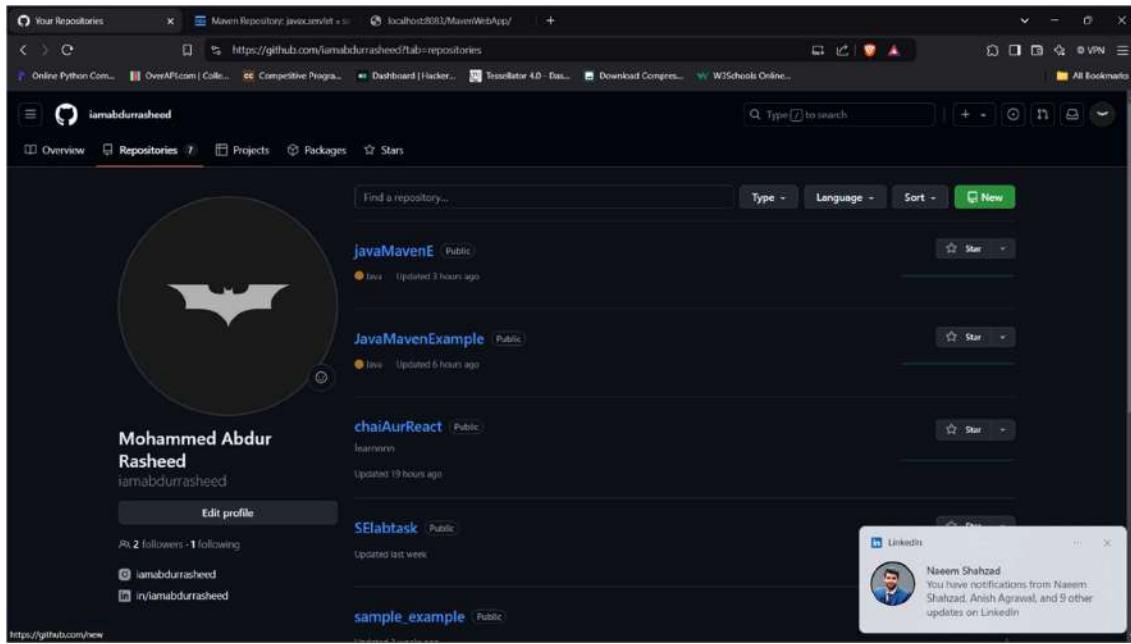


The preview/Output of our project is shown on localhost ..

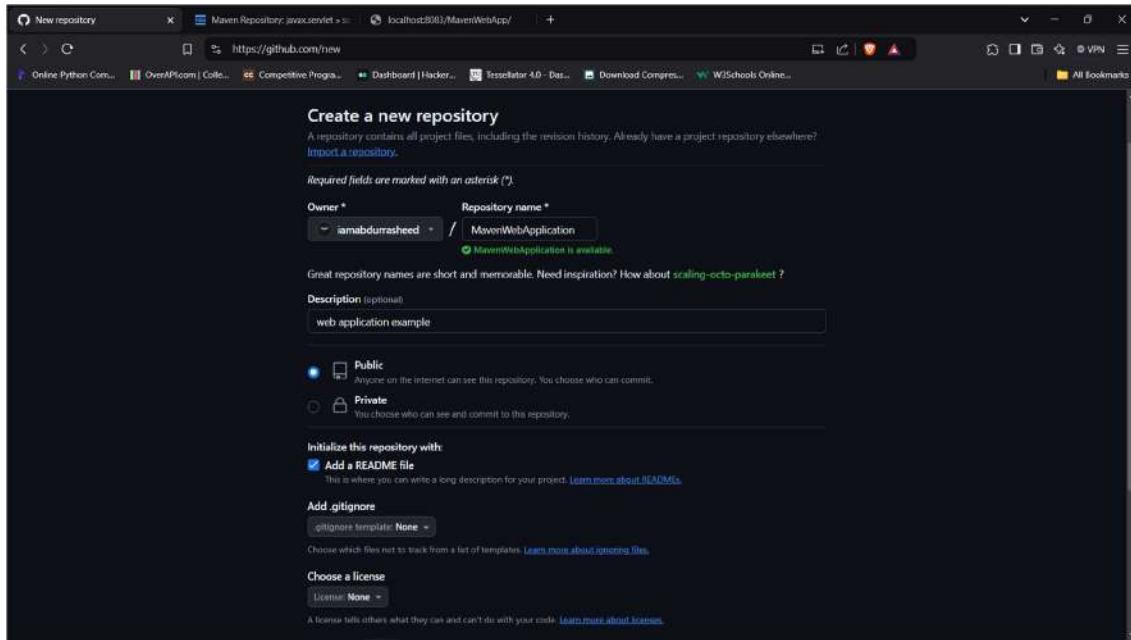


Next is to push the project into GitHub, for this, create a new repository in GitHub

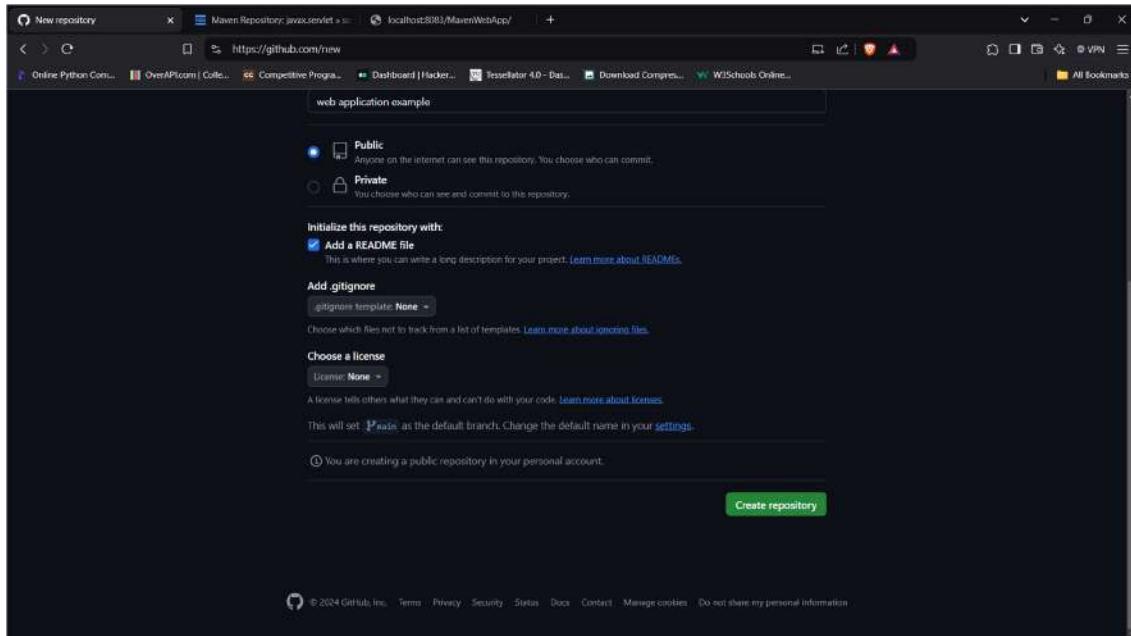
Click on New to create a new repo



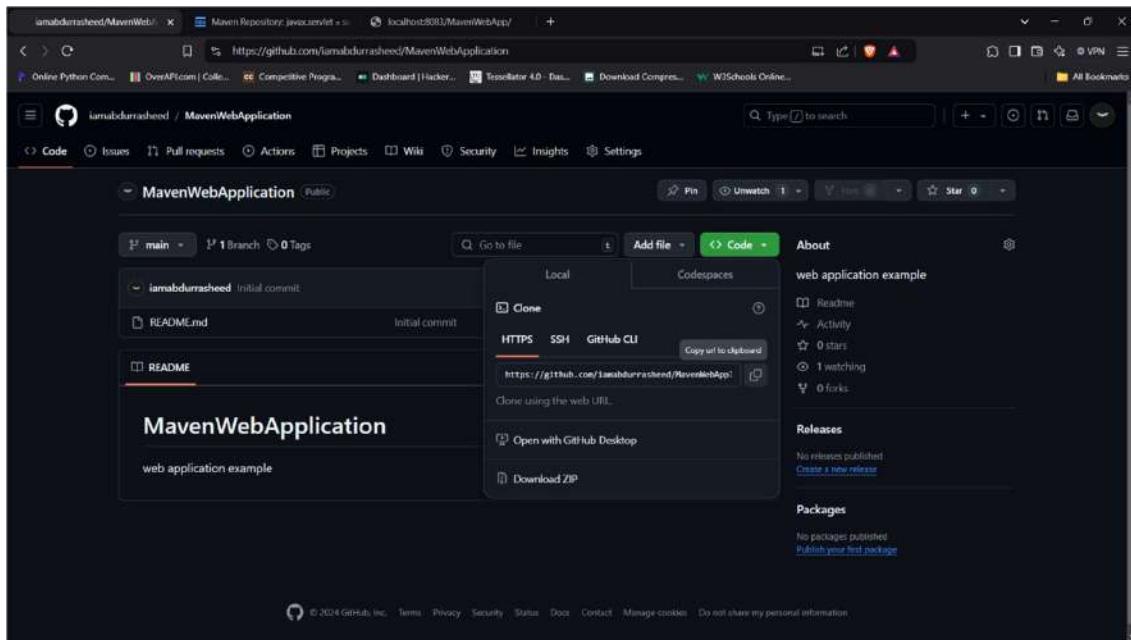
Enter the required fields



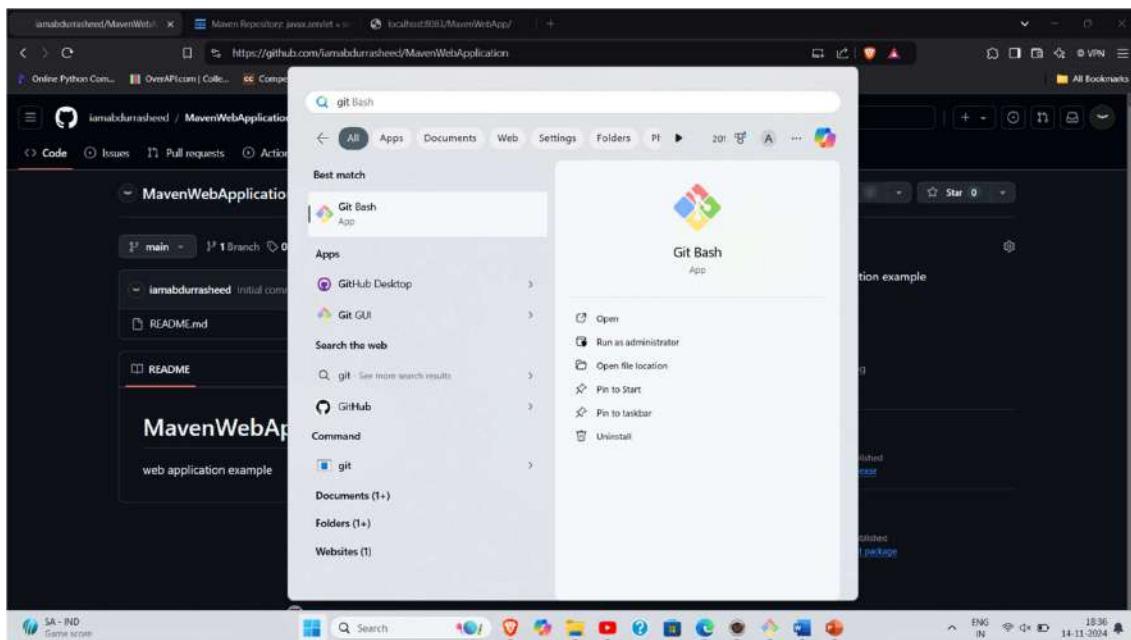
Click on Create repository



New repo is created ..copy the HTTPS key to clone



Open git bash



Clone the repository

```

MINGW64:/c/Users/abdur
$ git clone https://github.com/iamabdurrasheed/MavenwebApplication.git
Cloning into 'MavenwebApplication'...
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.
$ |

```

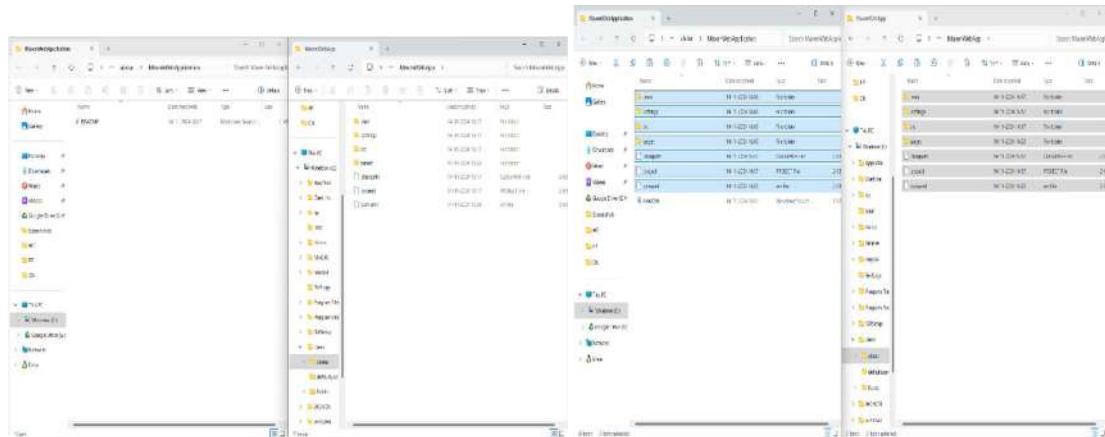
Change directory to the webApplication directory

```

MINGW64:/c/Users/abdur/MavenWebApplication
$ git clone https://github.com/iamabdurrasheed/MavenwebApplication.git
Cloning into 'MavenwebApplication'...
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.
$ cd MavenwebApplication
$ |

```

One way of cloning the local and global projects is to copy the contents from eclipse workspace to the local cloned repository as shown



Using Git bash, add and commit the changes made as shown
then using

```

MINGW4::c:\Users\abdul\MavenWebApplication
$ git add .
warning: in the working copy of 'src/main/webapp/WEB-INF/web.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/webapp/index.jsp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/MavenWebApp/WEB-INF/web.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/MavenWebApp/index.jsp', LF will be replaced by CRLF the next time Git touches it

ZANABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/MavenWebApplication (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/MavenWebApp.war
    new file: target/MavenWebApp/WEB-INF/lib/servlet-api-2.5.jar
    new file: target/MavenWebApp/WEB-INF/web.xml
    new file: target/MavenWebApp/index.jsp
    new file: target/maven-archiver/pom.properties

ZANABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/MavenWebApplication (main)
$ 

```

Commit the stages added

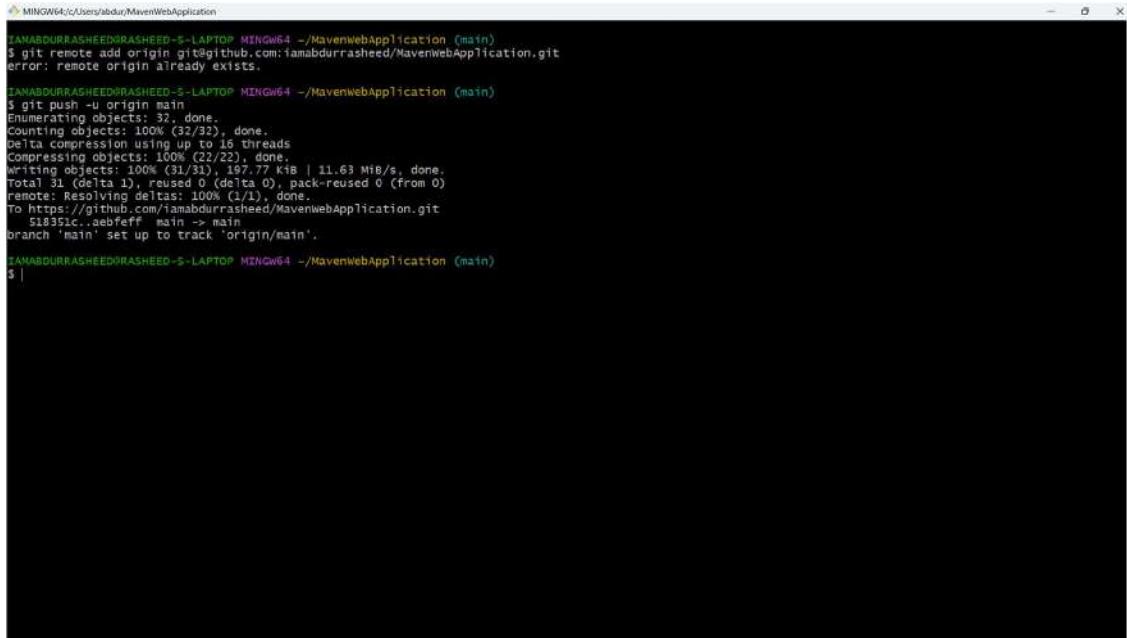
```

MINGW4::c:\Users\abdul\MavenWebApplication
$ git commit -m "committed webApp"
[main abefff] committed webApp
 21 files changed, 235 insertions(+)
create mode 100640 .classpath
create mode 100640 .mvn/jvm.config
create mode 100640 .mvn/maven.config
create mode 100640 .project
create mode 100640 .settings/.jsdtscope
create mode 100640 .settings/org.eclipse.core.resources.prefs
create mode 100640 .settings/org.eclipse.jdt.core.prefs
create mode 100640 .settings/org.eclipse.m2e.core.prefs
create mode 100640 .settings/org.eclipse.wst.common.component
create mode 100640 .settings/org.eclipse.wst.common.project.facet.core.xml
create mode 100640 .settings/org.eclipse.wst.jsdt.ui.superType.container
create mode 100640 .settings/org.eclipse.wst.jsdt.ui.superType.name
create mode 100640 .settings/org.eclipse.wst.validation.prefs
create mode 100640 pom.xml
create mode 100640 src/main/webapp/WEB-INF/web.xml
create mode 100640 src/main/webapp/index.jsp
create mode 100640 target/MavenWebApp.war
create mode 100640 target/MavenWebApp/WEB-INF/lib/servlet-api-2.5.jar
create mode 100640 target/MavenWebApp/WEB-INF/web.xml
create mode 100640 target/MavenWebApp/index.jsp
create mode 100640 target/maven-archiver/pom.properties

ZANABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/MavenWebApplication (main)
$ 

```

Then using Git remote add origin
 git push -u origin main
 push it into remote repository



```

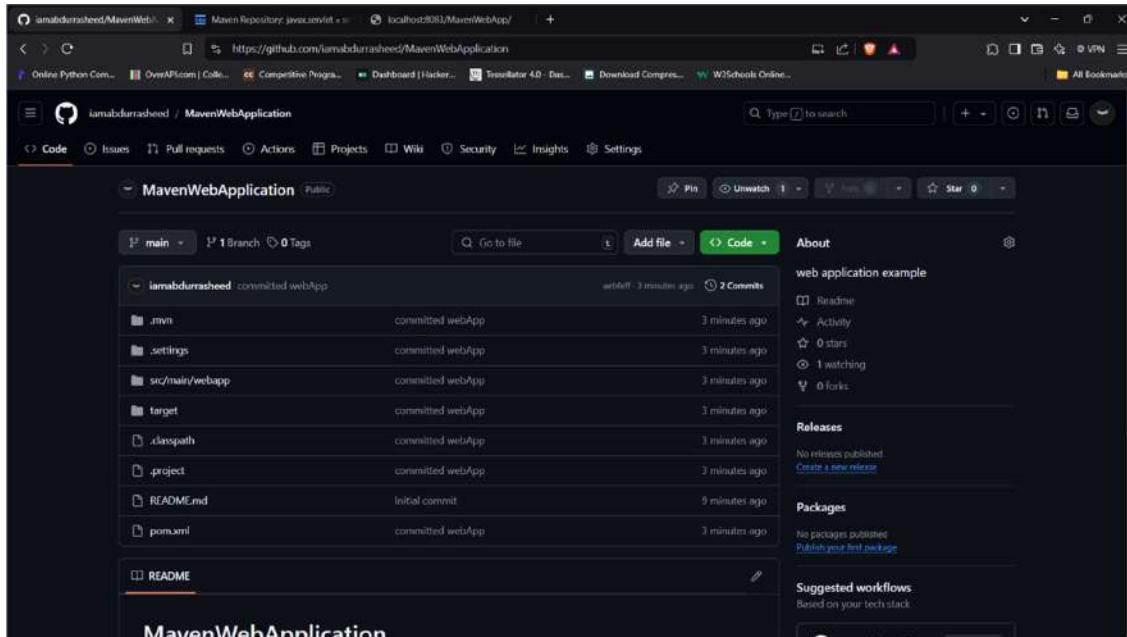
iAMABDURRASHEED@RASHEED-5-LAPTOP MINGW64 ~/MavenWebApplication (main)
$ git remote add origin git@github.com:iamabdurrasheed/MavenwebApplication.git
error: remote origin already exists.

iAMABDURRASHEED@RASHEED-5-LAPTOP MINGW64 ~/MavenWebApplication (main)
$ git push -u origin main
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 16 threads
Compressing objects: 100% (23/23), done.
Writing objects: 100% (31/31), 197.77 KiB | 11.63 MiB/s, done.
Total 31 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/iamabdurrasheed/MavenWebApplication.git
  51835ic..aebfeff main -> main
branch 'main' set up to track 'origin/main'.

iAMABDURRASHEED@RASHEED-5-LAPTOP MINGW64 ~/MavenWebApplication (main)
$ 

```

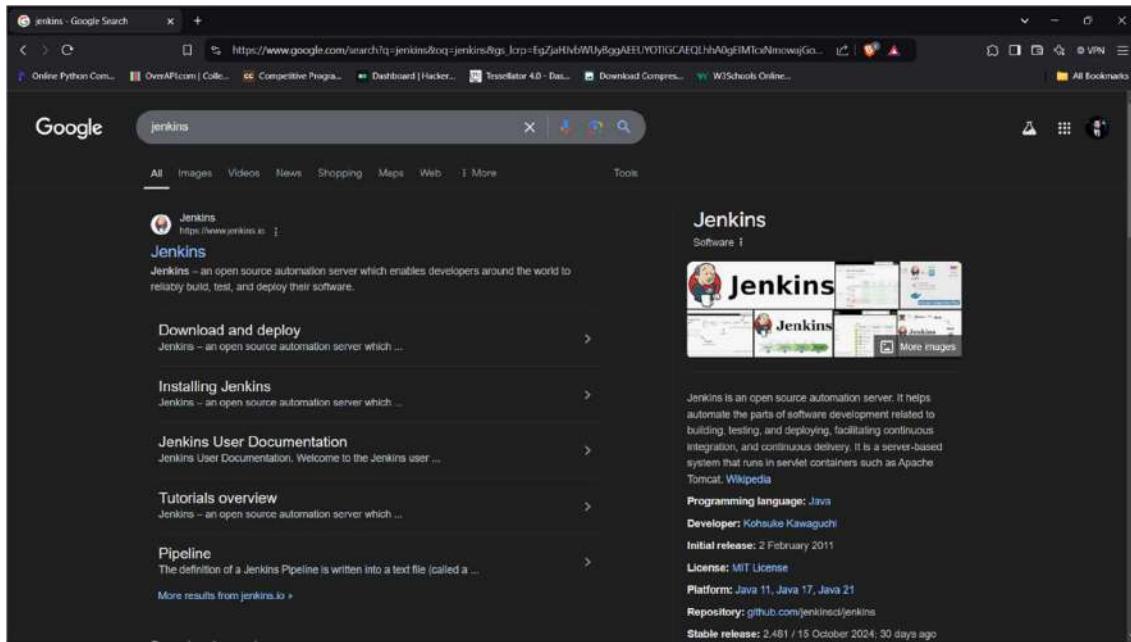
Refresh the repository



Succesfully Pushed into Remote Repository

5 C. INSTALLATION OF JENKINS

Download Jenkins from its official site



Jenkins official site



Scroll down to download and deploy

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...](#)

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

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

Downloading Jenkins

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

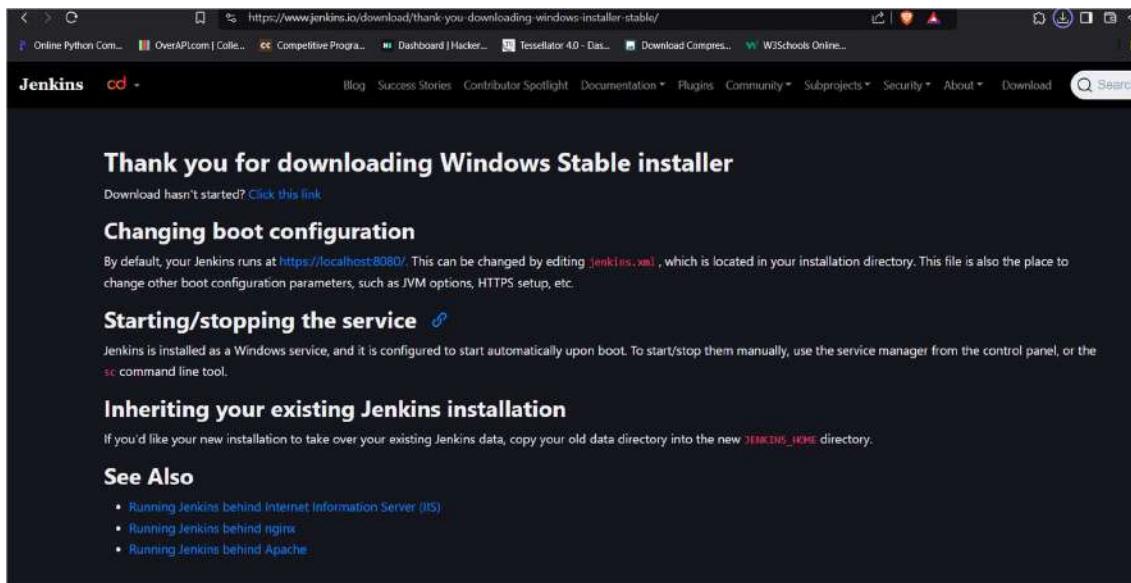
1. Before downloading, please take a moment to review the [Hardware and Software requirements](#) section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the [Installing Jenkins](#) section of the User Handbook.

Click on windows application to download.

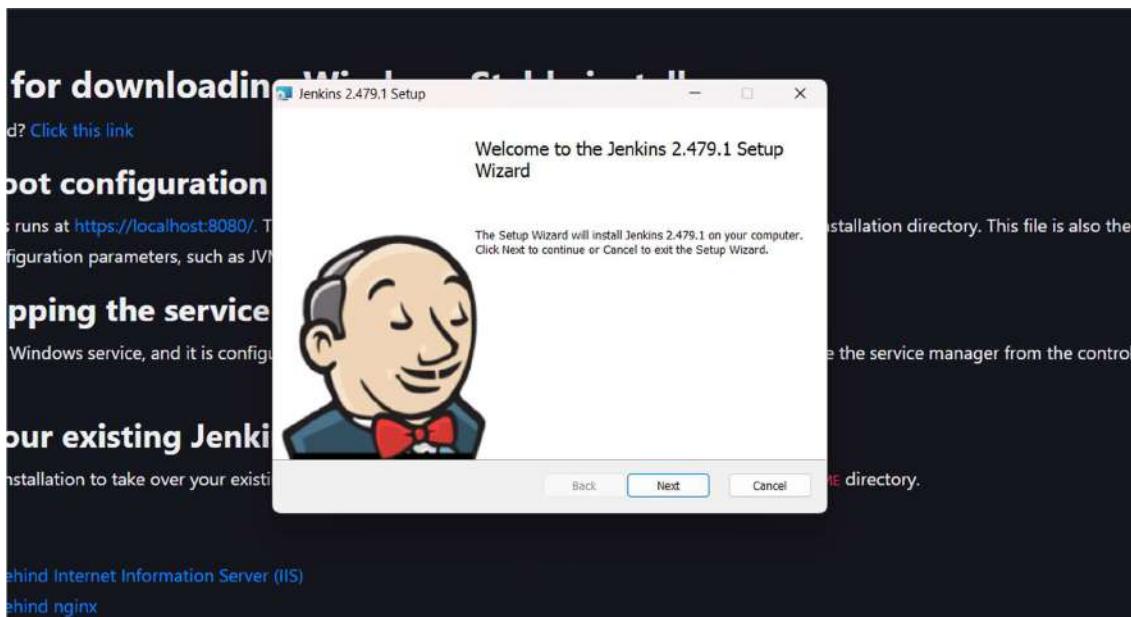
Download Jenkins 2.479.1 LTS for:

Platform	Status
Generic Java package (.war)	
Docker	
Kubernetes	
Ubuntu/Debian	
Red Hat/Fedora/Alma/Rocky/CentOS	
Windows	
openSUSE	
FreeBSD	Third party
Gentoo	Third party
macOS	Third party
OpenBSD	Third party

Wait until it installs

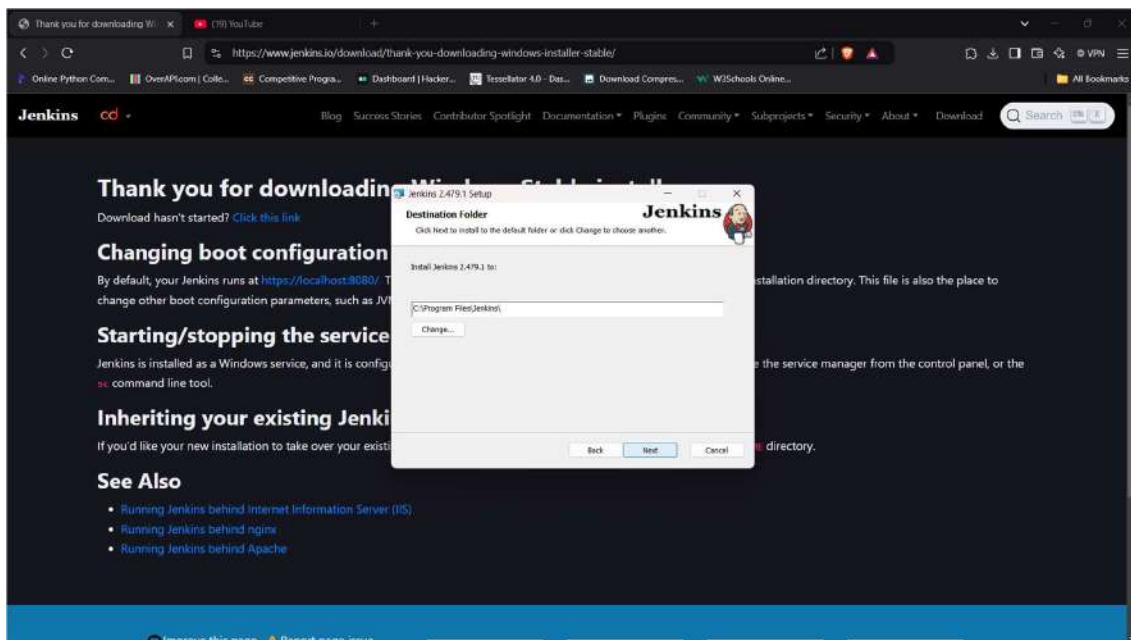


Open the installed folder

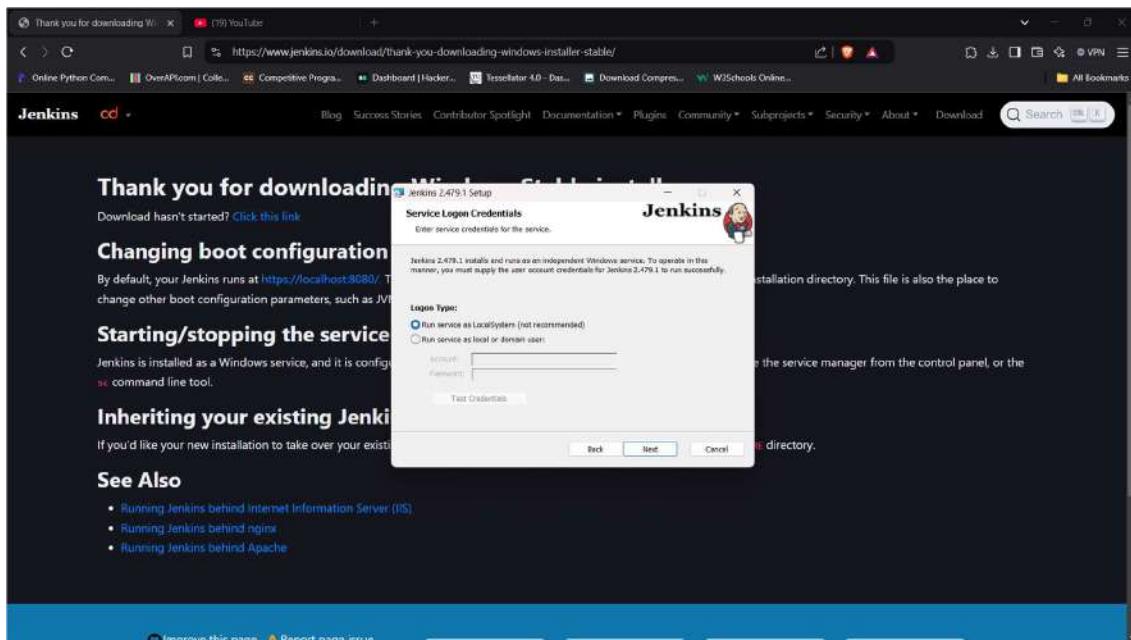


Click on Next

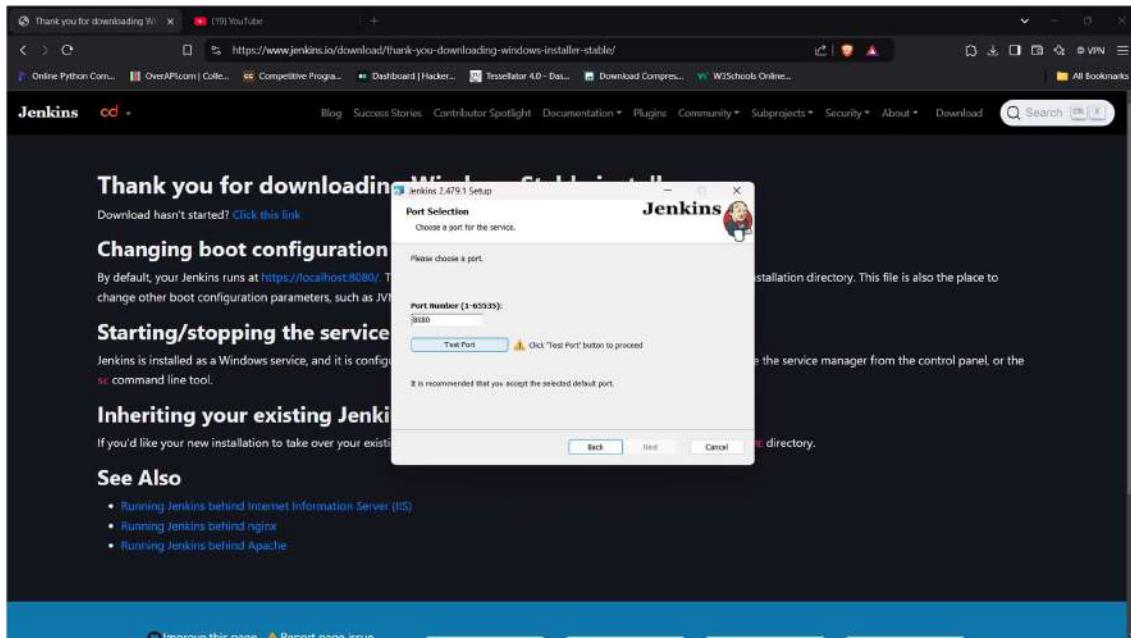
Click on Next



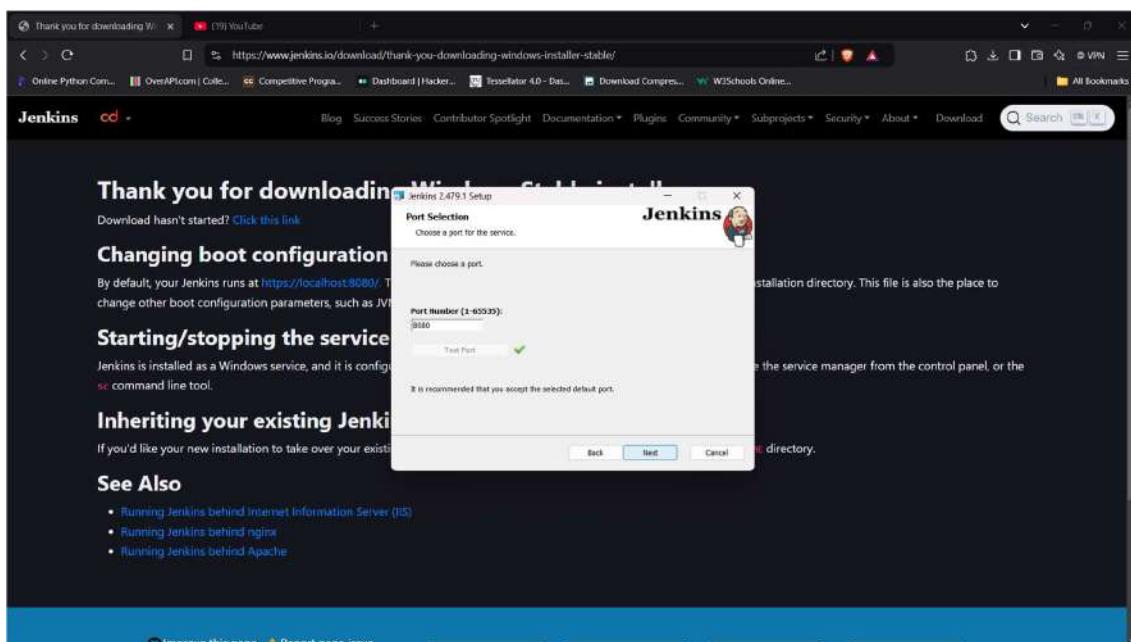
Click on Next



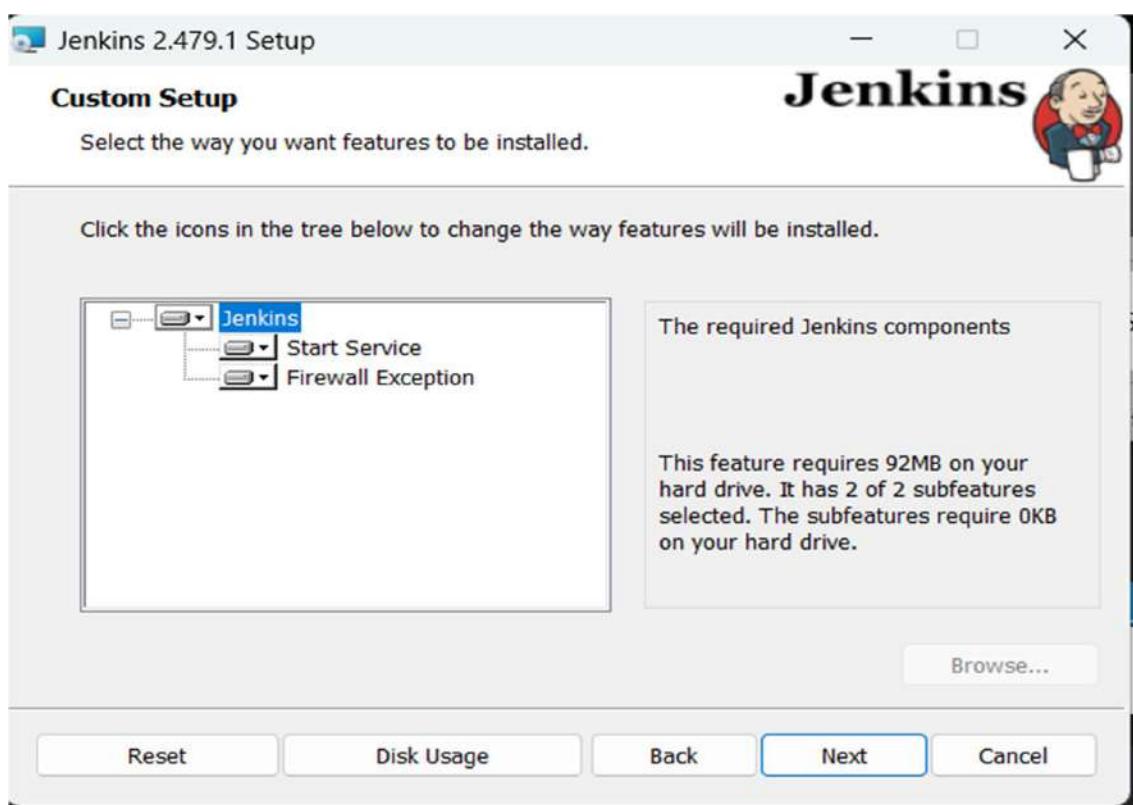
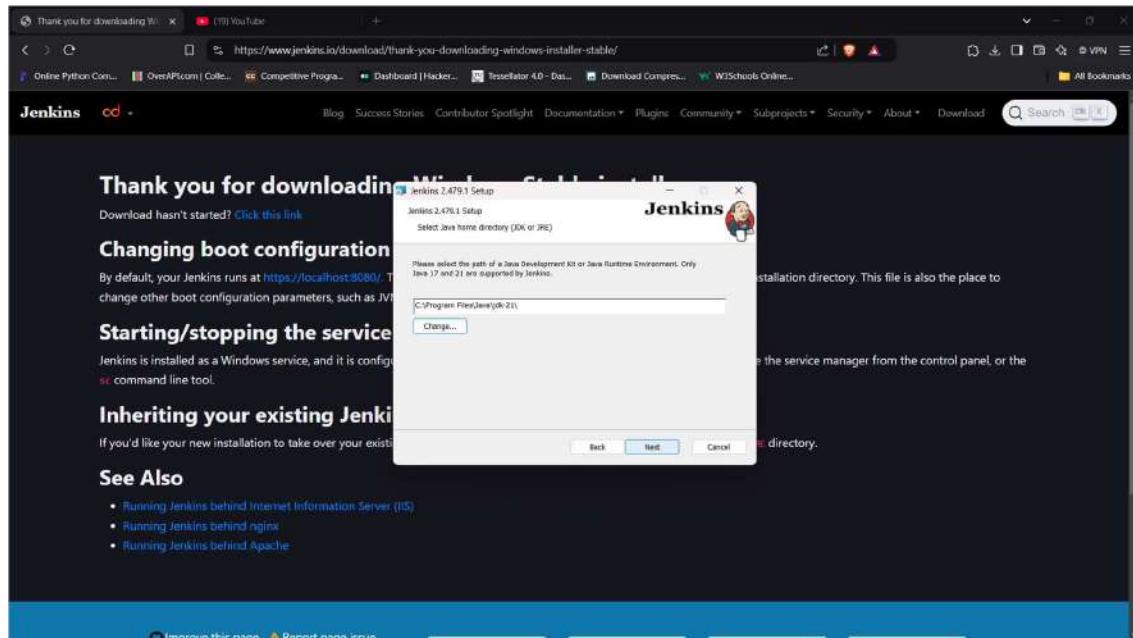
Click on Test Port



Click on Next



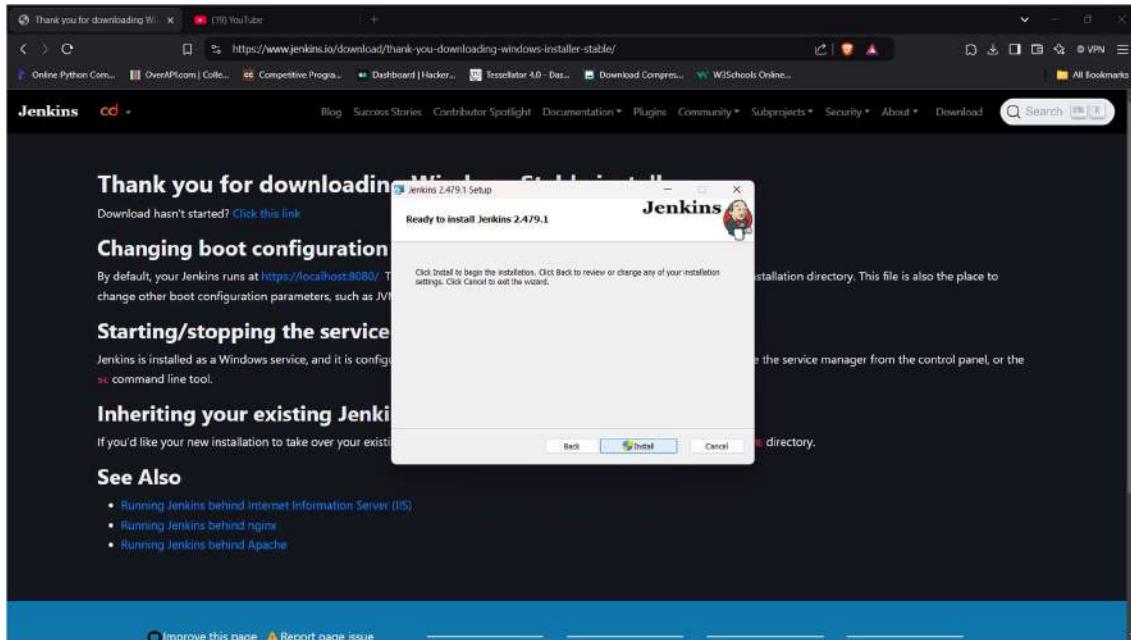
Click on Next



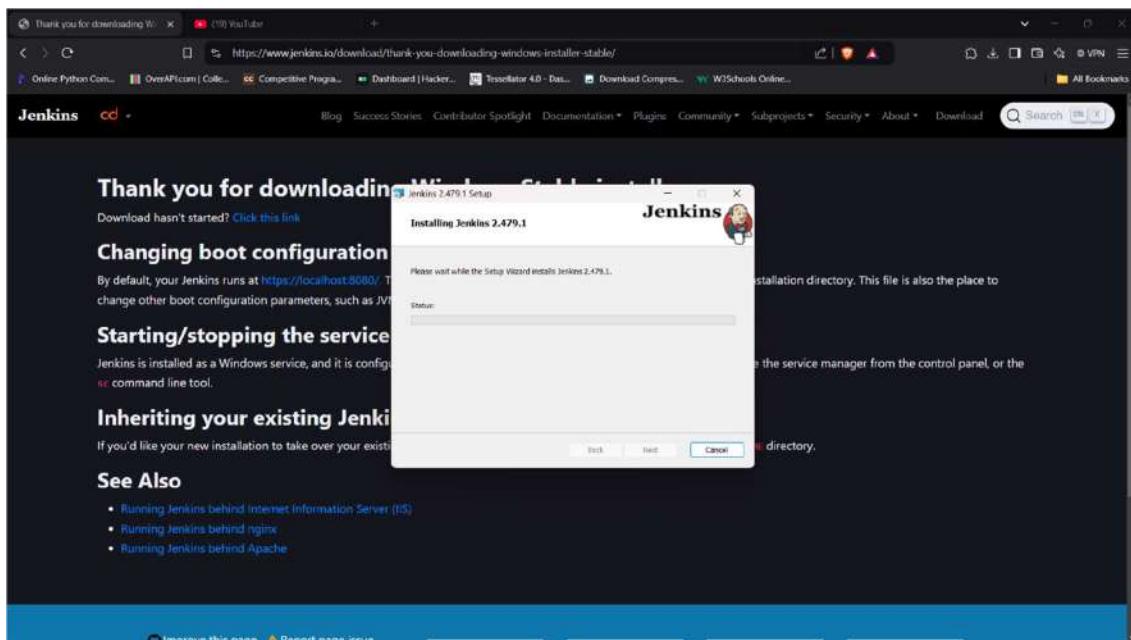
Click on Next

Click on Install

Click on Install



Wait for it to install



Click on Finish

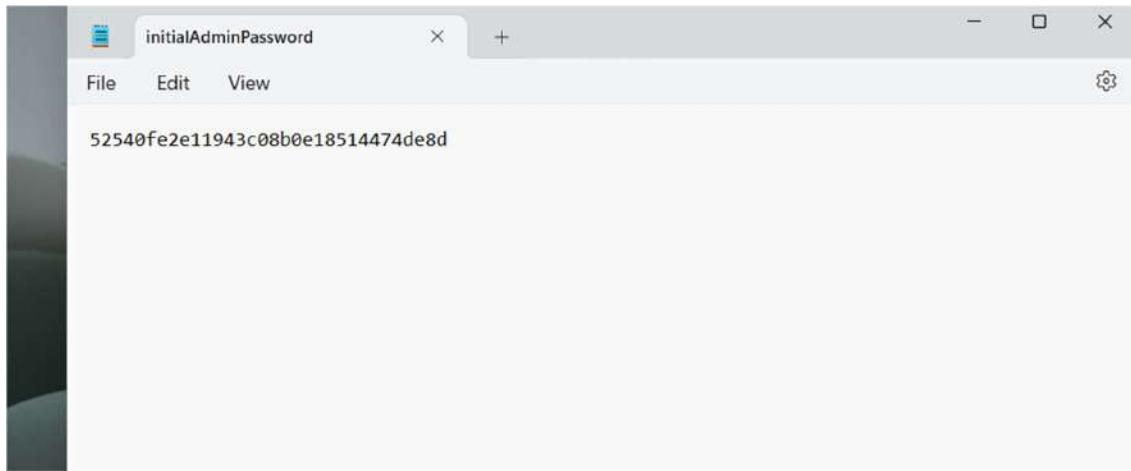


In browser open localhost:8080/jenkins

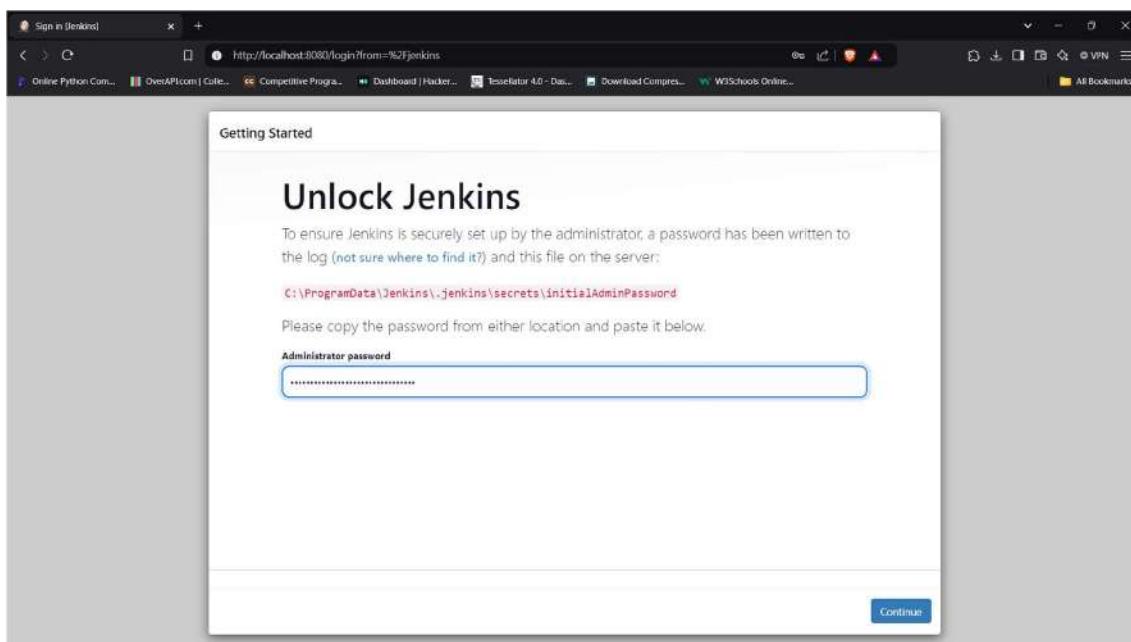
A screenshot of a web browser window. The address bar shows the URL "http://localhost:8080/login?from=%2fjenkins". The main content area has a header "Getting Started" and a large heading "Unlock Jenkins". Below the heading, there is text explaining that a password has been written to the log and a specific file path is provided: "C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword". It instructs the user to copy the password from either location and paste it into a text input field labeled "Administrator password". At the bottom right of the form is a blue "Continue" button.

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

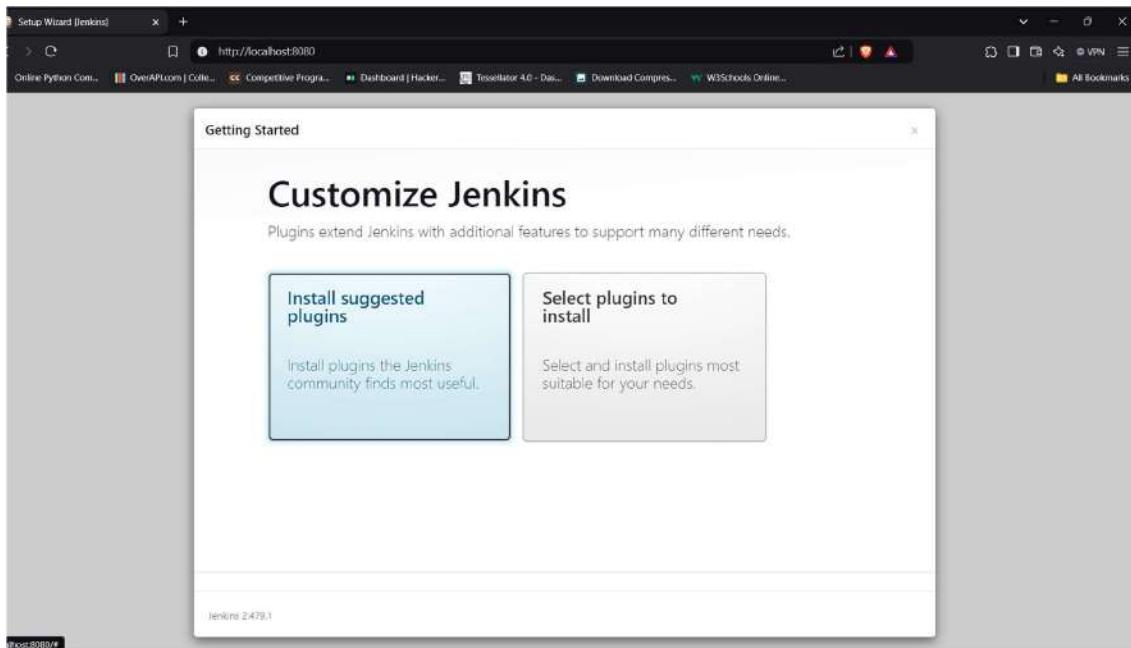
Paste the path in file manager copy the password



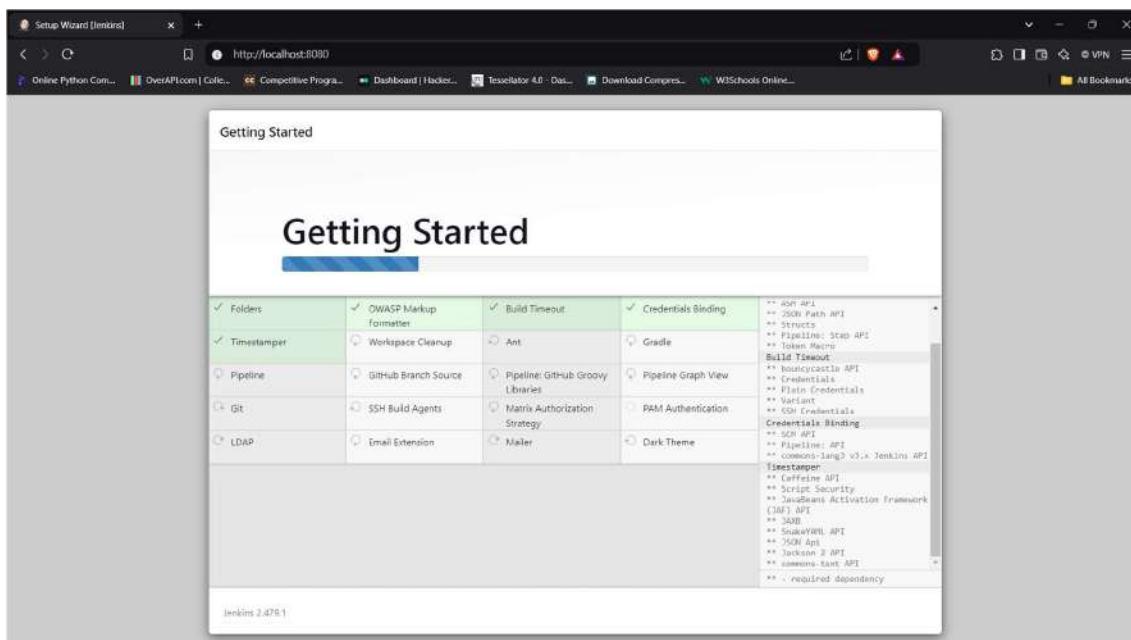
Unlock Jenkins using th password



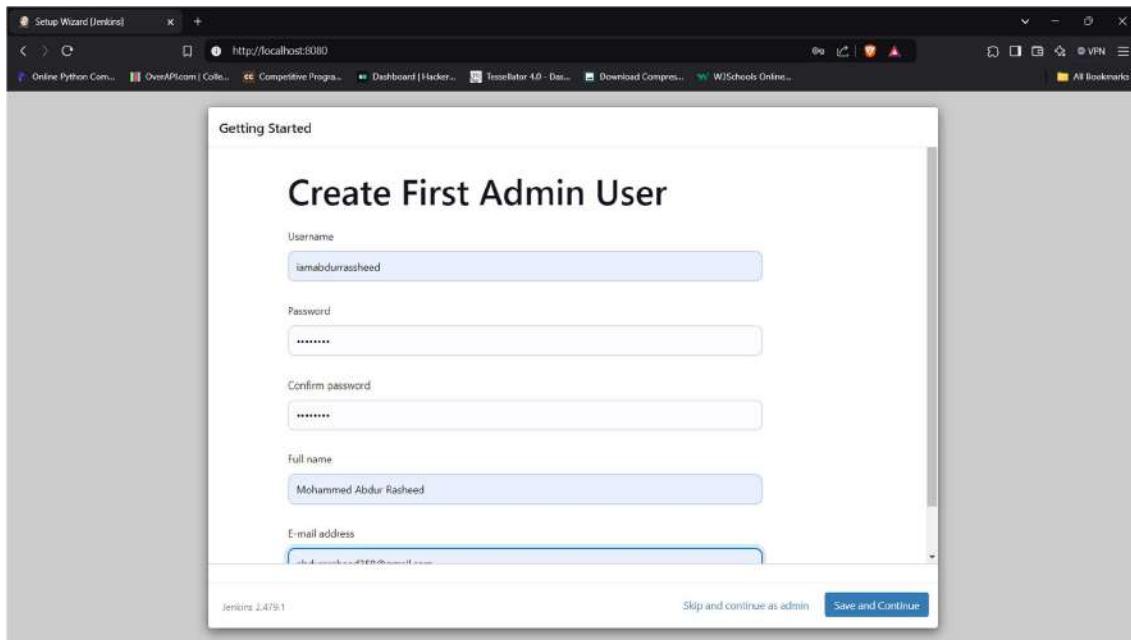
Follow as shown



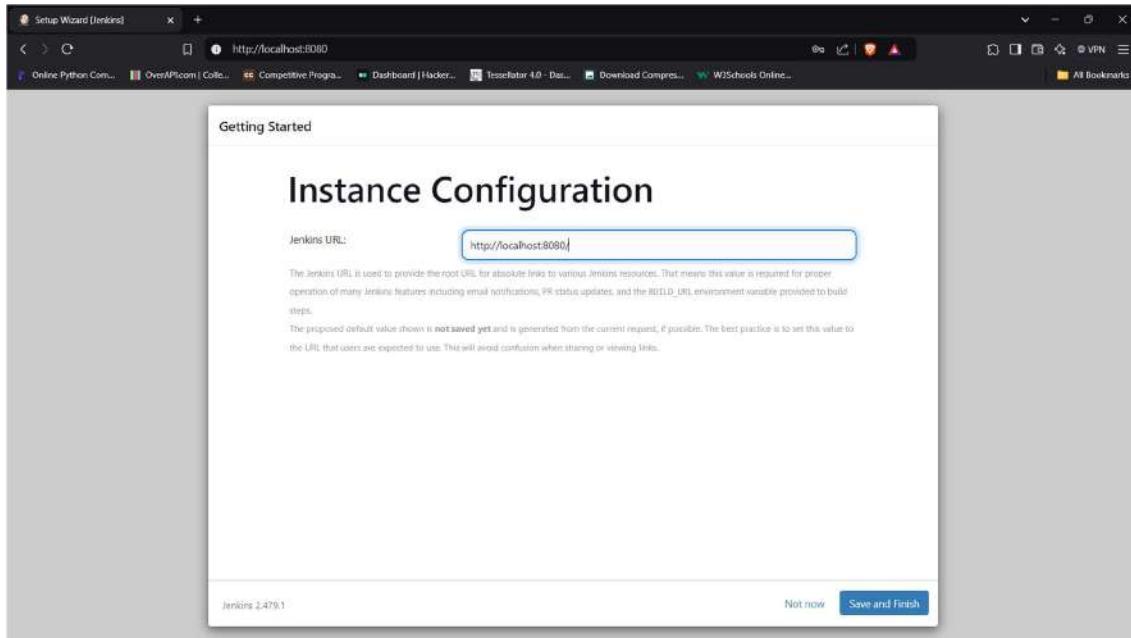
Wait until it gets started



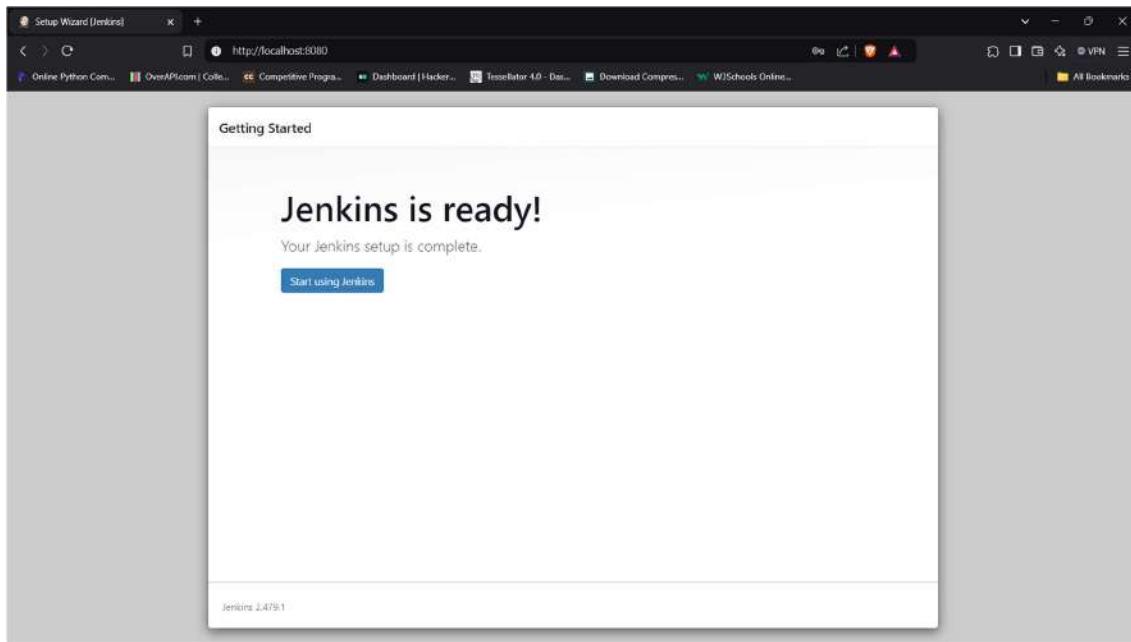
Create a user



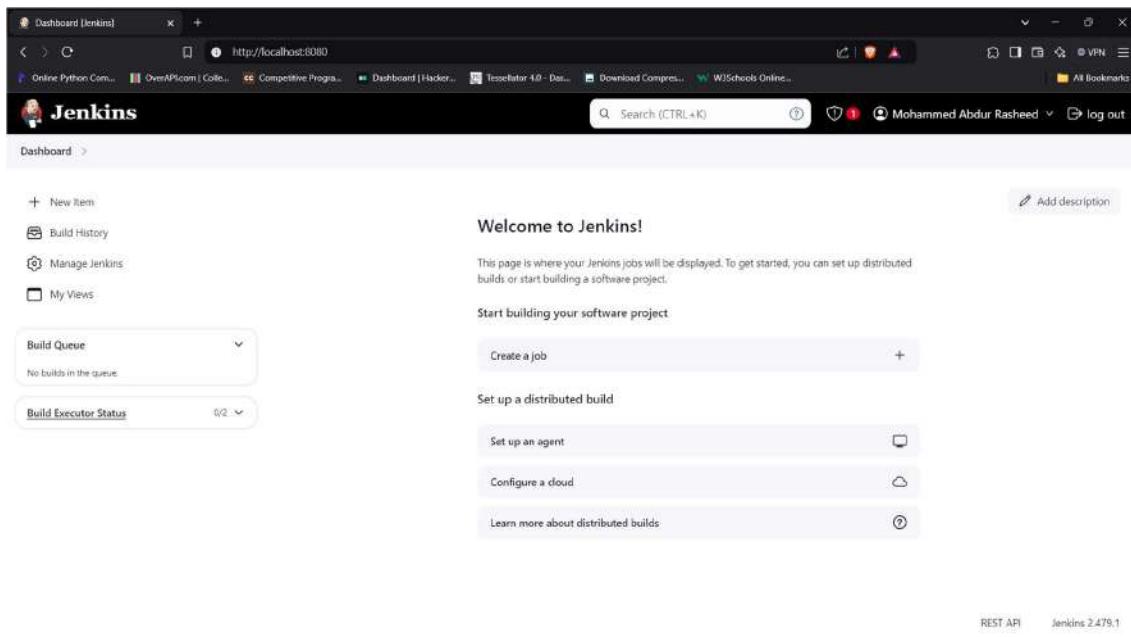
Click on save and finish



Click on start using jenkins



Jenkins default port must be used to access Jenkins



Manage Jenkins

The screenshot shows the Jenkins 'Manage Jenkins' page under the 'System Configuration' section. It includes sections for System, Tools, Plugins, Nodes, Clouds, Appearance, Security, Credentials, and Credential Providers. A message at the top right encourages setting up distributed builds. Buttons for 'Set up agent', 'Set up cloud', and 'Dismiss' are visible.

Manage Jenkins → Plugins → Available plugins

install the below listed plugins by searching them..

The screenshot shows the Jenkins 'Available plugins' page. It lists several plugins: Maven Integration 3.24, Pipeline Utility Steps 2.18.0, Copy Artifact 757-v05365585a_455, Deploy to container 1.16, and Build Pipeline 2.02. Each plugin entry includes its name, version, category, a brief description, and the date it was last updated.

Plugin	Version	Category	Last Updated
Maven Integration	3.24	Build Tools	24 days ago
Pipeline Utility Steps	2.18.0	pipeline, Build Tools, Miscellaneous	1 mo 6 days ago
Copy Artifact	757-v05365585a_455	Build Parameters, Build Tools	23 days ago
Deploy to container	1.16	Artifact Uploader	4 yr 0 mo ago
Build Pipeline	2.02	User Interface, Build Tools, Other Post-Build Actions	

Wait for them to get installed

Dashboard > Manage Jenkins > Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

Plugin	Status
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
Loading plugin extensions	Success
Javadoc	Success
iSch dependency	Success
Maven Integration	Success
Commons Compress API	Installing
Pipeline Utility Steps	Pending
Copy Artifact	Pending
JavaMail API	Pending
SSH server	Pending
Deploy to container	Pending
Parameterized Trigger	Pending
jQuery	Pending
Build Pipeline	Pending
Loading plugin extensions	Pending

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

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

localhost:8080/manage/pluginManager/ REST API Jenkins 2.479.1

Now restart Jenkins by clicking on the below check box

Dashboard > Manage Jenkins > Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

Plugin	Status
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
Loading plugin extensions	Success
Javadoc	Success
iSch dependency	Success
Maven Integration	Success
Commons Compress API	Success
Pipeline Utility Steps	Success
Copy Artifact	Success
JavaMail API	Success
SSH server	Success
Deploy to container	Success
Parameterized Trigger	Success
jQuery	Success
Build Pipeline	Success
Loading plugin extensions	Success

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

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

localhost:8080/manage/pluginManager/ REST API Jenkins 2.479.1

Wait for it to restart

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links for New Item, Build History, Manage Jenkins (which is selected), and My Views. Below the sidebar are sections for Build Queue (0 builds in the queue) and Build Executor Status (0/2). The main area is titled "Manage Jenkins" and contains a message about building on the built-in node being a security issue. It also lists a warning for the "Build Pipeline Plugin 2.0.2" which has a "Stored XSS vulnerability (no fix available)". There are buttons to Go to plugin manager and Configure which of these warnings are shown. The "System Configuration" section includes links for System, Tools, Plugins, Nodes, Clouds, and Appearance.

Manage Jenkins → Tools

The screenshot shows the Jenkins Tools configuration page. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins > Tools. The main content area is titled "Tools". It includes sections for "Maven Configuration" (with fields for Default settings provider and Use default maven settings), "JDK installations" (with an "Add JDK" button), and "Git installations" (with a table showing one entry for "Git" with a "Name" field, "Save" button, and "Apply" button).

Configuring Java

The screenshot shows the Jenkins configuration interface for adding a Java JDK installation. The 'JDK installations' section is open, and a new 'JDK' entry is being configured. The 'Name' field contains 'JAVA_HOME'. The 'JAVA_HOME' path is set to 'C:\Program Files\Java\jdk-21'. A checkbox for 'Install automatically' is unchecked. At the bottom of the dialog are 'Save' and 'Apply' buttons.

Configuring git

The screenshot shows the Jenkins configuration interface for adding a Git installation. The 'Git installations' section is open, and a new 'Git' entry is being configured. The 'Name' field contains 'GIT_HOME'. The 'Path to Git executable' is set to 'C:\Program Files\Git\bin\git.exe'. A checkbox for 'Install automatically' is unchecked. At the bottom of the dialog are 'Save' and 'Apply' buttons.

Configuring MAVEN

The screenshot shows the Jenkins configuration interface for adding a Maven installation. The 'Gradle installations' section is open, and a new 'Maven' entry is being configured. The 'Name' field contains 'MAVEN_HOME'. The 'MAVEN_HOME' path is set to 'C:\Users\abdun\OneDrive\Desktop\apache-maven-3.9.9-bin\apache-maven-3.9.9'. A checkbox for 'Install automatically' is unchecked. At the bottom of the dialog are 'Save' and 'Apply' buttons.

Jenkins 2.479.1

follow as shown

Apply → Save

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

Create a new item build for maven java project

New Item

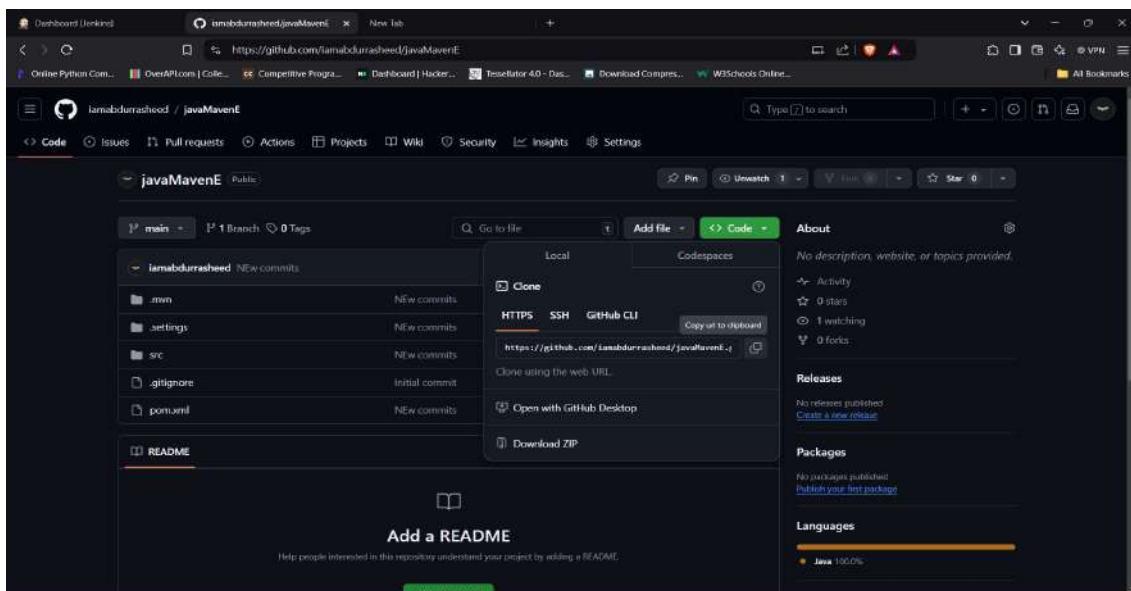
Enter an item name

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**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Get the repository of our java project from github



Enter the git http url

Dashboard > PresentProject_build > Configuration

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

Credentials [?](#)

+ Add

Advanced [▼](#)

Add Repository

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

Save **Apply**

Copy the github link of project

Branch is the main

The screenshot shows a build configuration interface. At the top, there's a breadcrumb navigation: Dashboard > PresentProject_build > Configuration. Below this, there are two unchecked checkboxes: "GitHub hook trigger for GITScm polling" and "Poll SCM". A dropdown menu is open, listing several build steps:

- Copy artifacts from another project
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit
- Trigger/call builds on other projects

The "Invoke top-level Maven targets" option is highlighted with a light blue background. Below the dropdown, there's a button labeled "Add build step ^".

Under the "Post-build Actions" section, there's a button labeled "Add post-build action ▾".

At the bottom, there are two buttons: "Save" (in a blue rounded rectangle) and "Apply".

Add build steps → invoke-top-level maven targets

The screenshot shows the Jenkins configuration interface for a project named "PresentProject_build". The "Configuration" section is open, displaying two "Build Steps".

- Build Step 1:** "Invoke top-level Maven targets" (Goals: clean)
- Build Step 2:** "Invoke top-level Maven targets" (Goals: install)

At the bottom of the configuration page are "Save" and "Apply" buttons.

Setup the maven targets to clean

The screenshot shows the Jenkins configuration interface for a project named "PresentProject_build". The "Configuration" section is open, and a dropdown menu is displayed under the "Post-build Actions" heading.

- Archive the artifacts
- Build other projects
- Publish JUnit test result report
- Publish Javadoc
- Record fingerprints of files to track usage
- Git Publisher
- Build other projects (manual step)
- Deploy war/ear to a container
- E-mail Notification
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Trigger parameterized build on other projects
- Delete workspace when build is done
- Add post-build action ^

At the bottom of the configuration page are "Save" and "Apply" buttons.

Add post-build action → Archive the artifacts

The screenshot shows the Jenkins configuration interface for a project named 'PresentProject_build'. In the 'Post-build Actions' section, there is a single action named 'Archive the artifacts'. The 'Files to archive' field contains the pattern '**/*'. Below the action are 'Save' and 'Apply' buttons. At the bottom right, it says 'REST API Jenkins 2.479.1'.

Setup the maven targets to install

Now to archive all files write **/* then write the other project to build after it

The screenshot shows the Jenkins configuration interface for a project named 'PresentProject_build'. A dropdown menu is open under 'Add post-build action', and the option 'Build other projects' is selected. Other options in the menu include 'Aggregate downstream test results', 'Archive the artifacts', 'Publish JUNIT test result report', 'Publish Javadoc', 'Record fingerprints of files to track usage', 'Git Publisher', 'Build other projects (manual step)', 'Deploy war/ear to a container', 'E-mail Notification', 'Editable Email Notification', 'Set GitHub commit status (universal)', 'Set build status on GitHub commit [deprecated]', 'Trigger parameterized build on other projects', and 'Delete workspace when build is done'. Below the dropdown are 'Save' and 'Apply' buttons. At the bottom right, it says 'REST API Jenkins 2.479.1'.

Archive the artifacts

Files to archive ?
**/*

Build other projects

Projects to build
PresentProject_test

No such project 'PresentProject_test'. Did you mean 'PresentProject_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**

REST API Jenkins 2.479.1

Add post-build action → Build other projects

Now create a test smaple and configure it as follows

New Item

Enter an item name
PresentProject_test

Select an item type

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

OK

Check and follow to setup configuration

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 ?

Build Environment

- Delete workspace before build starts
 - Advanced ▾
 - Use secret text(s) or file(s) ?
 - Add timestamps to the Console Output
 - Inspect build log for published build scans
 - Terminate a build if it's stuck
 - With Ant ?

Build Steps

Save Apply

Configure

General Source Code Management Build Triggers Build Environment **Build Steps** Post-build Actions

Copy artifacts from another project

Project name ?
sampleMavenJava_build

Which build ?
Latest successful build

Stable build only

Artifacts to copy ?
**/*

Artifacts not to copy ?

Now set the maven targets to test

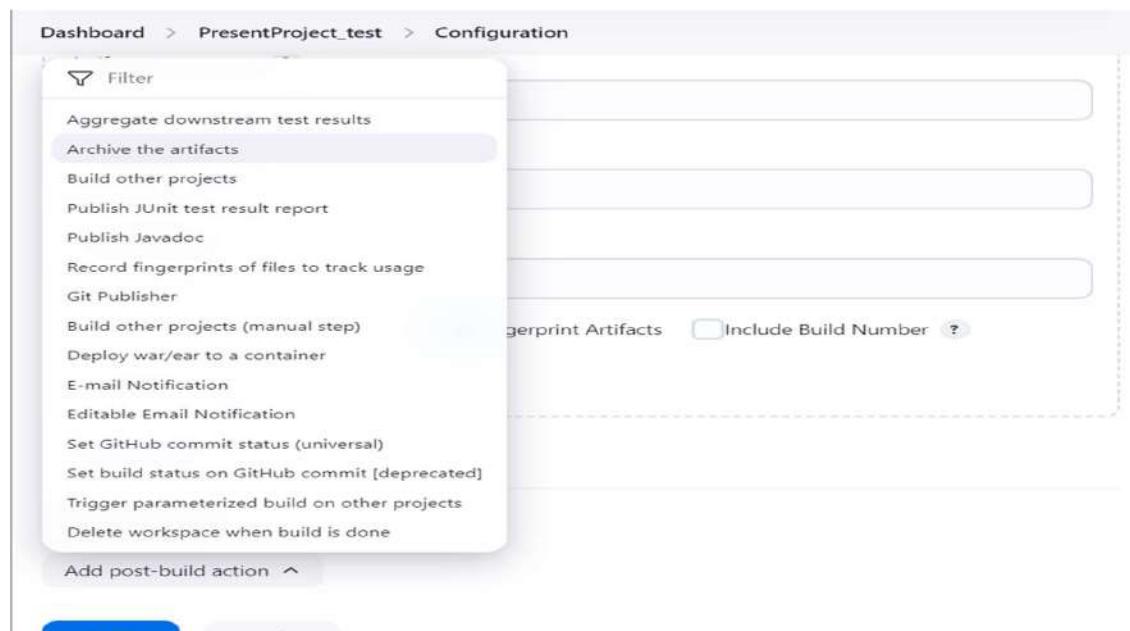
Now copy artifacts from the previous build

The screenshot shows the Jenkins 'Configuration' screen for a project named 'PresentProject_test'. At the top, there are several build triggers: 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. Below these, a dropdown menu is open under the heading 'Copy artifacts from another project'. The menu lists various build steps: 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', 'Set build status to "pending" on GitHub commit', and 'Trigger/call builds on other projects'. At the bottom of the dropdown is a link 'Add build step ^'. Below the dropdown, there is a section titled 'Post-build Actions' with a button 'Add post-build action ^'. At the bottom of the configuration screen are 'Save' and 'Apply' buttons.

Enter test project details as showed below

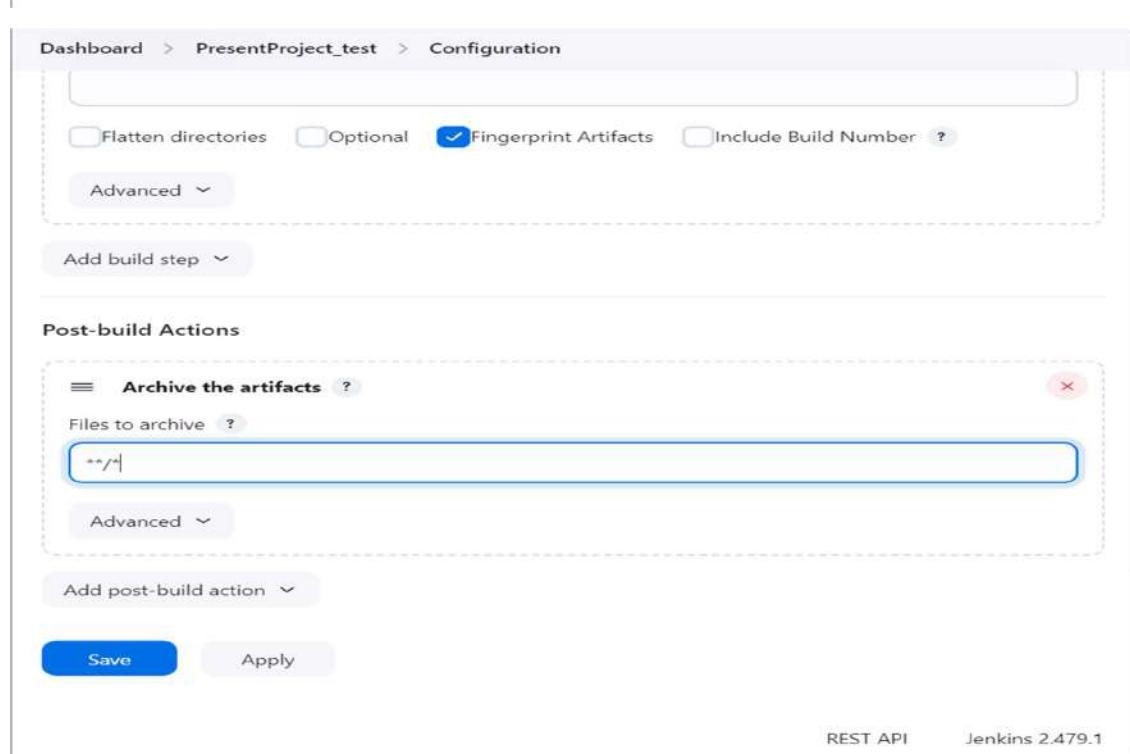
The screenshot shows the 'Copy artifacts from another project' configuration dialog. It includes fields for 'Project name' (with a note about a missing project), 'Which build' (set to 'Latest successful build'), 'Stable build only' (checkbox checked), 'Artifacts to copy' (set to '**/*'), 'Artifacts not to copy' (empty), 'Target directory' (empty), and 'Parameter filters' (empty). At the bottom are 'Save' and 'Apply' buttons.

Archive all the artifacts and save apply



The screenshot shows the Jenkins configuration page for the project 'PresentProject_test'. The 'Archive the artifacts' step is highlighted in the left sidebar. On the right, there are fields for 'Fingerprint Artifacts' and 'Include Build Number'. At the bottom are 'Save' and 'Apply' buttons.

REST API Jenkins 2.479.1



The screenshot shows the Jenkins configuration page for the project 'PresentProject_test'. The 'Archive the artifacts' step is now applied, as indicated by the checked checkbox in the sidebar. The 'Fingerprint Artifacts' and 'Include Build Number' options are also present. At the bottom are 'Save' and 'Apply' buttons.

REST API Jenkins 2.479.1

Apply → Save

Create a new view as a pipeline

New view

Name: PresentProject_pipeline

Type:

- Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.
- List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.
- My View
This view automatically displays all the jobs that the current user has an access to.

Create

REST API Jenkins 2.479.1

Select the initial job i.e build project

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: PresentProject_build

Trigger Options

Build Cards: Standard build card

Use the default build cards

Restrict triggers to most recent successful builds: No

Always allow manual trigger on pipeline steps

OK **Apply**

Click on Apply→Ok

Dashboard > PresentProject_pipeline > Configure

Row Headers

Just the pipeline number

Show just the build pipeline number

Column Headers

No header

Do not show any column headers

Refresh frequency (in seconds) ?

3

URL for custom CSS files

Console Output Link Style

Lightbox

OK **Apply**

REST API Jenkins 2.479.1

In pipeline we can see the project is built successfully

Search (CTRL+K)

Mohammed Abdur Rasheed

Dashboard > PresentProject_pipeline >

Build Pipeline

Run History Configure Add Step Delete Manage

Pipeline #1

#1 PresentProject_build

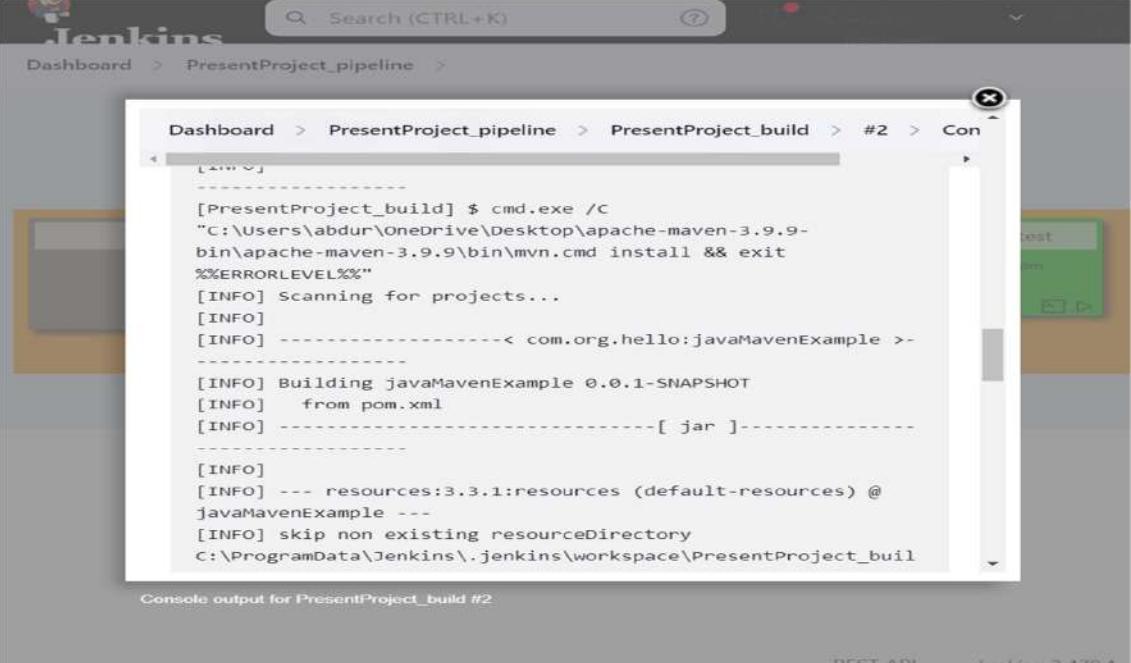
- 24-Nov-2024 2:46:35 pm
- 52 sec
- iamabdurrassheed

#1 PresentProject_test

- 24-Nov-2024 2:49:35 pm
- 1.2 sec

REST API Jenkins 2.479.1

#BuildNO→Console(for build)



The screenshot shows the Jenkins interface with the URL [http://localhost:8080/job/PresentProject_pipeline/job/PresentProject_build/2/console](#). The console output window displays the following Maven build logs:

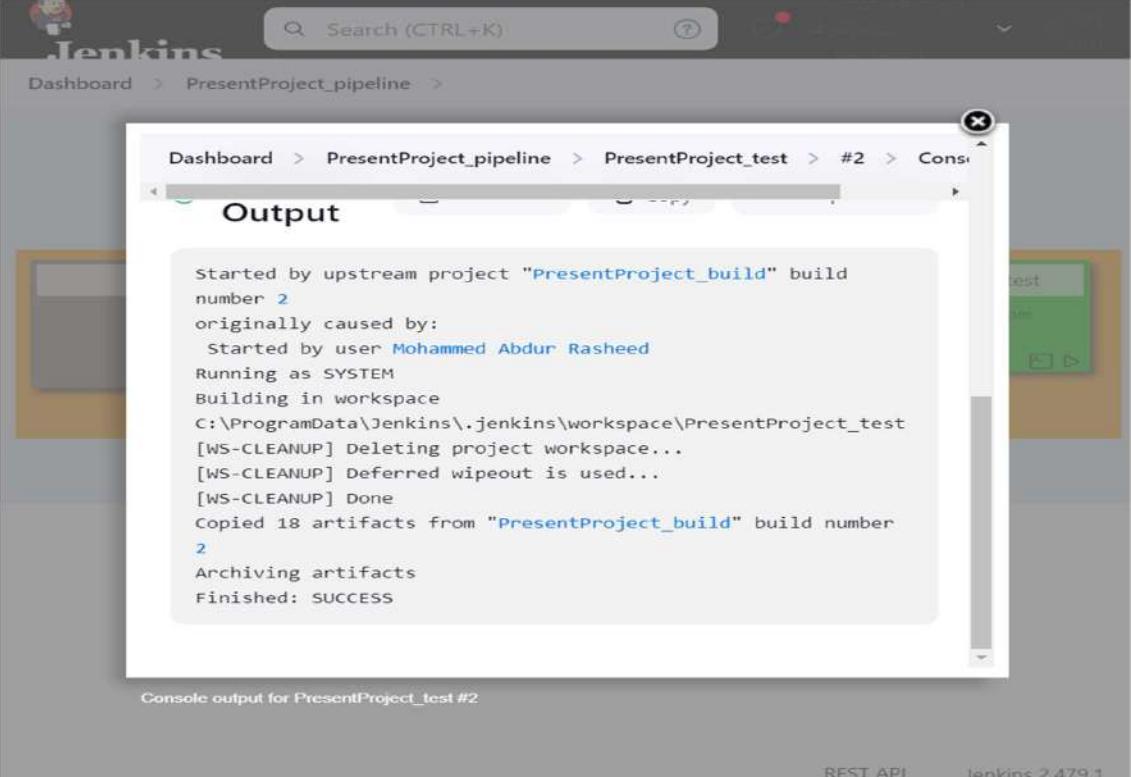
```

[PresentProject_build] $ cmd.exe /C
"C:\Users\abdur\OneDrive\Desktop\apache-maven-3.9.9-bin\apache-maven-3.9.9\bin\mvn.cmd install && exit %ERRORLEVEL%"'
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.org.hello:javAMavenExample >-----
[INFO] Building javAMavenExample 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @
javAMavenExample ---
[INFO] skip non existing resourceDirectory
C:\ProgramData\Jenkins\.jenkins\workspace\PresentProject_build

```

Console output for PresentProject_build #2

#BuildNO→Console(for test)



The screenshot shows the Jenkins interface with the URL [http://localhost:8080/job/PresentProject_pipeline/job/PresentProject_test/2/console](#). The console output window is titled "Output" and displays the following log entries:

```

Started by upstream project "PresentProject_build" build
number 2
originally caused by:
Started by user Mohammed Abdur Rasheed
Running as SYSTEM
Building in workspace
C:\ProgramData\Jenkins\.jenkins\workspace\PresentProject_test
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] Done
Copied 18 artifacts from "PresentProject_build" build number
2
Archiving artifacts
Finished: SUCCESS

```

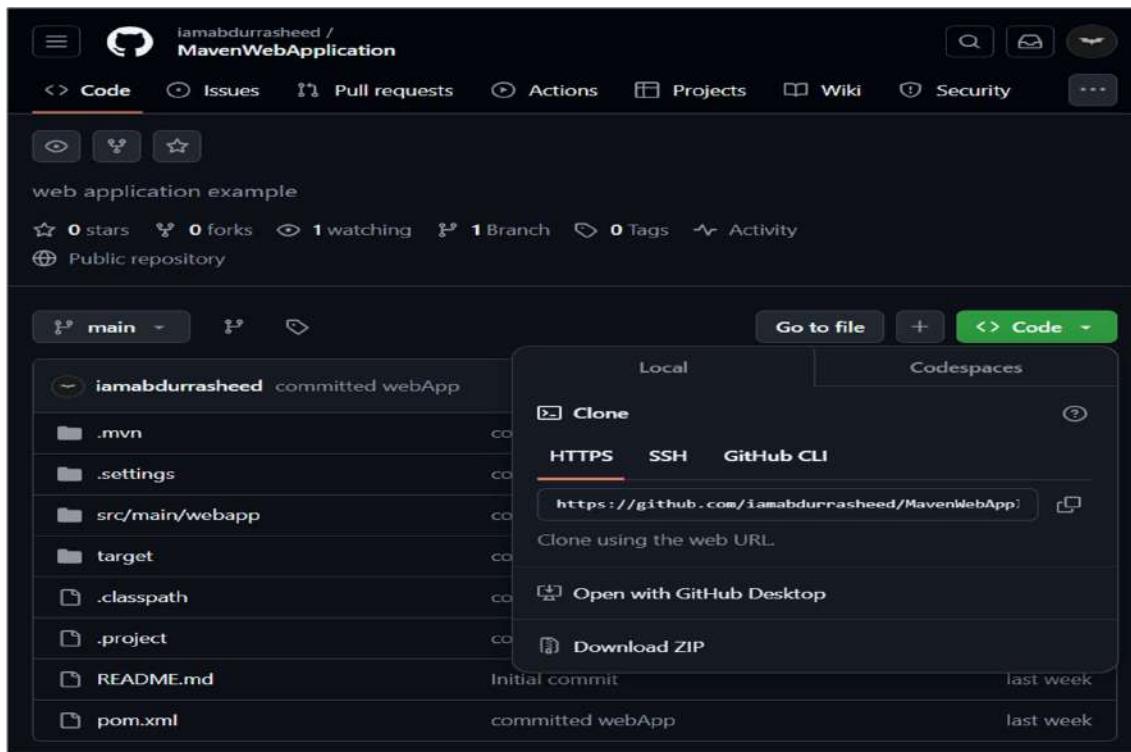
Console output for PresentProject_test #2

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

Create a new item for sample web and select freestyle

The screenshot shows the Jenkins interface for creating a new item. At the top, there's a navigation bar with a search bar, a help icon, and a user profile for 'Mohammed Abdur Rasheed'. Below the navigation bar, the path 'Dashboard > All > New Item' is visible. The main title 'New Item' is displayed, followed by the instruction 'Enter an item name' and a text input field containing 'sampleMavenWeb_build'. Under 'Select an item type', there are four options: 'Freestyle project' (selected), 'Pipeline', 'Maven project', and 'Multi-configuration project'. Each option has a brief description and a corresponding icon. At the bottom left is a blue 'OK' button.

Go to github and copy the link of your web project



Paste it here and then keep branch as main

Dashboard > sampleMavenWeb_build > Configuration

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)
https://github.com/iamabdurrasheed/MavenWebApplication.git

Credentials [?](#)
- none -

+ Add

Advanced [?](#)

Add Repository

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)
*/master

Save Apply

then keep branch as main

Dashboard > sampleMavenWeb_build > Configuration

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Then in build triggers select poll SCM and add 5 stars to run every minute

Dashboard > sampleMavenWeb_build > Configuration

Additional Behaviours

Add ▾

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 Sunday, 24 November, 3:43:28 pm India Standard Time; would next run at Sunday, 24 November, 2024, 3:43:28 pm India Standard Time.

Ignore post-commit hooks ?

Save Apply

Build Steps → Invoke top-level Maven targets
goals:clean & install

Dashboard > sampleMavenWeb_build > Configuration

Build Steps

- Invoke top-level Maven targets** ?
- Maven Version: MAVEN_HOME
- Goals: clean
- Advanced

- Invoke top-level Maven targets** ?
- Maven Version: MAVEN_HOME
- Goals: install
- Advanced

Save **Apply**

Archive the artifacts by **/*

Dashboard > sampleMavenWeb_build > Configuration

Goals

- install
- Advanced

Add build step ▾

Post-build Actions

- Archive the artifacts** ?
- Files to archive: ?
- **/*
- Advanced

Add post-build action ▾

Save **Apply**

Build other project test apply and save

The screenshot shows the Jenkins configuration interface for a job named 'sampleMavenWeb_build'. In the 'Build other projects' section, there is an error message: 'No such project 'sampleMavenWeb_test'. Did you mean 'sampleMavenWeb_build''? Below the message 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'. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

Create a maven web test by clicking new item

The screenshot shows the Jenkins 'New Item' creation page. The user has entered 'sampleMavenWeb_test' as the item name. Under 'Select an item type', the 'Freestyle project' option is selected, described as a 'Classic, general-purpose job type'. Other options shown include 'Maven project', 'Pipeline', and 'Multi-configuration project'. At the bottom of the page is an 'OK' button.

Click OK

In source code management & Build Environment

Source Code Management

- None
- Git ?

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 ?

Build Environment

- Delete workspace before build starts
 - Advanced ▾
 - Use secret text(s) or file(s) ?
 - Add timestamps to the Console Output
 - Inspect build log for published build scans

31 Save Apply

None \

Delete workspace before build starts \

Copy Artifacts from other project

Build Steps

Copy artifacts from another project

Project name ?

sampleMavenWeb_build

Which build ?

Latest successful build

Stable build only

Artifacts to copy ?

**/*

Artifacts not to copy ?

Target directory ?

31 Save Apply

Add test for maven targets

Dashboard > sampleMavenWeb_test > Configuration

Advanced ▾

Invoke top-level Maven targets ?

Maven Version
MAVEN_HOME

Goals
test

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Save **Apply**

REST API Jenkins 2.479.1

Post build Actions → add deploy project to build

Dashboard > sampleMavenWeb_test > Configuration

Advanced ▾

Add build step ▾

Post-build Actions

Build other projects ?

Projects to build
sampleMavenWeb_deploy

No such project 'sampleMavenWeb_deploy'. Did you mean 'sampleMavenWeb_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**

REST API Jenkins 2.479.1

Next build the deploy project

Then apply and save

Now create a deploy item

The image shows two screenshots of the Jenkins interface. The top screenshot is titled 'Configuration' and includes sections for 'Source Code Management' (set to 'None'), 'Build Triggers' (with several options like 'Trigger builds remotely' and 'GitHub hook trigger for GITScm polling' available), and 'Build Environment' (with 'Delete workspace before build starts' checked). The bottom screenshot is titled 'Copy artifacts from another project' and shows fields for 'Project name' (set to 'sampleMavenWeb_test'), 'Which build' (set to 'Latest successful build'), and 'Artifacts to copy' (containing '**/*'). Both screenshots have 'Save' and 'Apply' buttons at the bottom.

Copy the artifacts of test item

Use war files

The screenshot shows two Jenkins configuration pages for a job named "sampleMavenWeb_deploy".

Configuration 1:

- Deploy war/ear to a container:**
 - WAR/EAR files:** `**/*.war`
 - Context path:** `examplewebproject`
 - Containers:**
 - Tomcat 9.x Remote:**
 - Credentials:** `admin/********`
 - Tomcat URL:** `http://localhost:8083/`
 - Advanced:** (dropdown menu)
- Buttons:** Save, Apply

Configuration 2:

- examplewebproject** (Container name)
- Containers:**
 - Tomcat 9.x Remote:**
 - Credentials:** `admin/********`
 - Tomcat URL:** `http://localhost:8083/`
 - Advanced:** (dropdown menu)
- Add Container:** (dropdown menu)
- Checkboxes:**
 - Deploy on failure
- Add post-build action:** (dropdown menu)
- Buttons:** Save, Apply

REST API Jenkins 2.479.1

Give the tomcat admin username and password

Then mention the port number of your server

Now apply and save the item

Now build the web project

sampleMavenWeb_build

Last Successful Artifacts

Downstream Projects

sampleMavenWeb_test

Permalinks

- Last build (#1), 18 min ago
- Last stable build (#1), 18 min ago
- Last successful build (#1), 18 min ago
- Last completed build (#1), 18 min ago

Builds

Filter: /

Today

#1 3:56PM

Open the pipeline and see the build of project

Build Pipeline

Run History Configure Add Step Delete Manage

Pipeline #15

#15 sampleMavenWeb_build

#15 sampleMavenWeb_test

#15 sampleMavenWeb_deploy

Finally the project is build, test and deployed

When we run our tomcat server we can see the page

The screenshot shows the Apache Tomcat 9.0.45 Manager interface at <http://localhost:8083/manager/html>. The title bar says "Tomcat Web Application Manager". The main content area displays a table of applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
MavenWebApp	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
exampleswebproject	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Below the table, there is a yellow "Deploy" button with the text "Deploy directory or WAR file located on server".

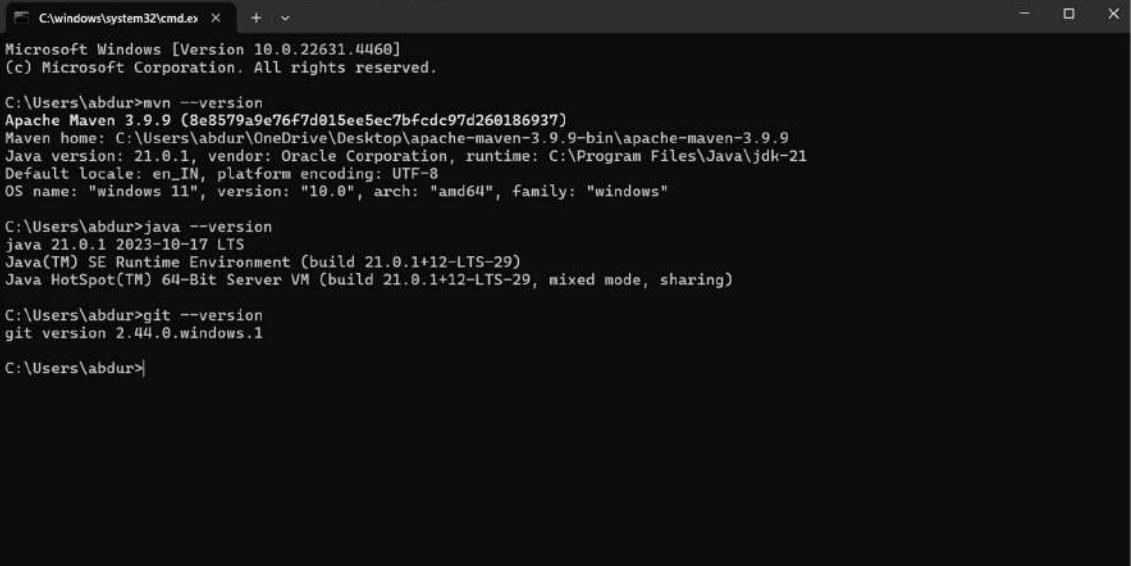
Tomcat web application manager

The screenshot shows a browser window at <http://localhost:8083/examplewebproject/>. The title bar says "localhost:8083/examplewebproject". The page content displays the text "Hello World!".

Finally when we go to the context path of our project we can see the page running

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

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



```
C:\windows\system32\cmd.exe > + .v
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abdur>mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfc97d260186937)
Maven home: C:\Users\abdur\OneDrive\Desktop\apache-maven-3.9.9-bin\apache-maven-3.9.9
Java version: 21.0.1, 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"

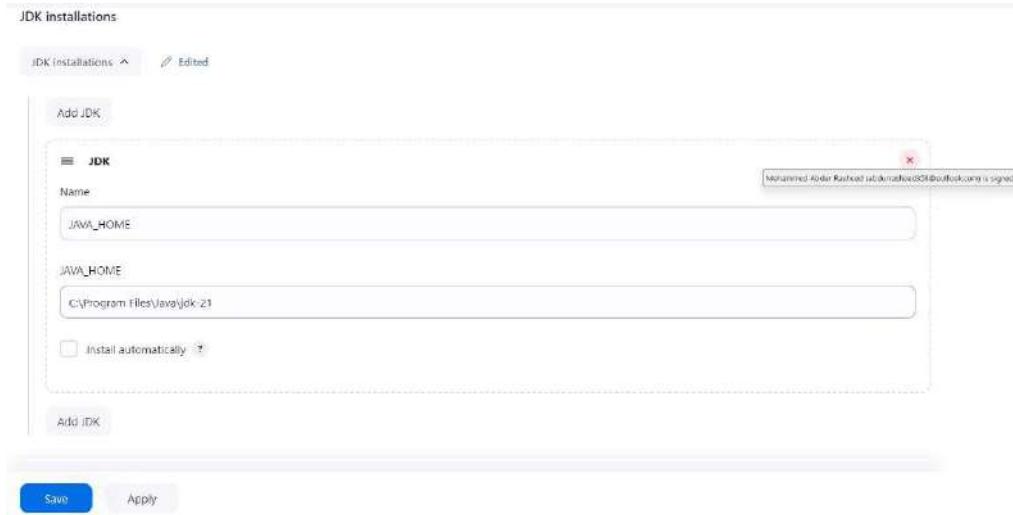
C:\Users\abdur>java --version
java 21.0.1 2023-10-17 LTS
Java(TM) SE Runtime Environment (build 21.0.1+12-LTS-29)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.1+12-LTS-29, mixed mode, sharing)

C:\Users\abdur>git --version
git version 2.44.0.windows.1

C:\Users\abdur>
```

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

Configuring JDK



Configuring Git

The screenshot shows the Jenkins configuration interface for adding a new Git installation. The 'Git installations' section is active, displaying fields for 'Name' (set to 'GIT_HOME'), 'Path to Git executable' (set to 'C:\Program Files\Git\bin\git.exe'), and an unchecked 'Install automatically' checkbox. Below this is an 'Add Git' button. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

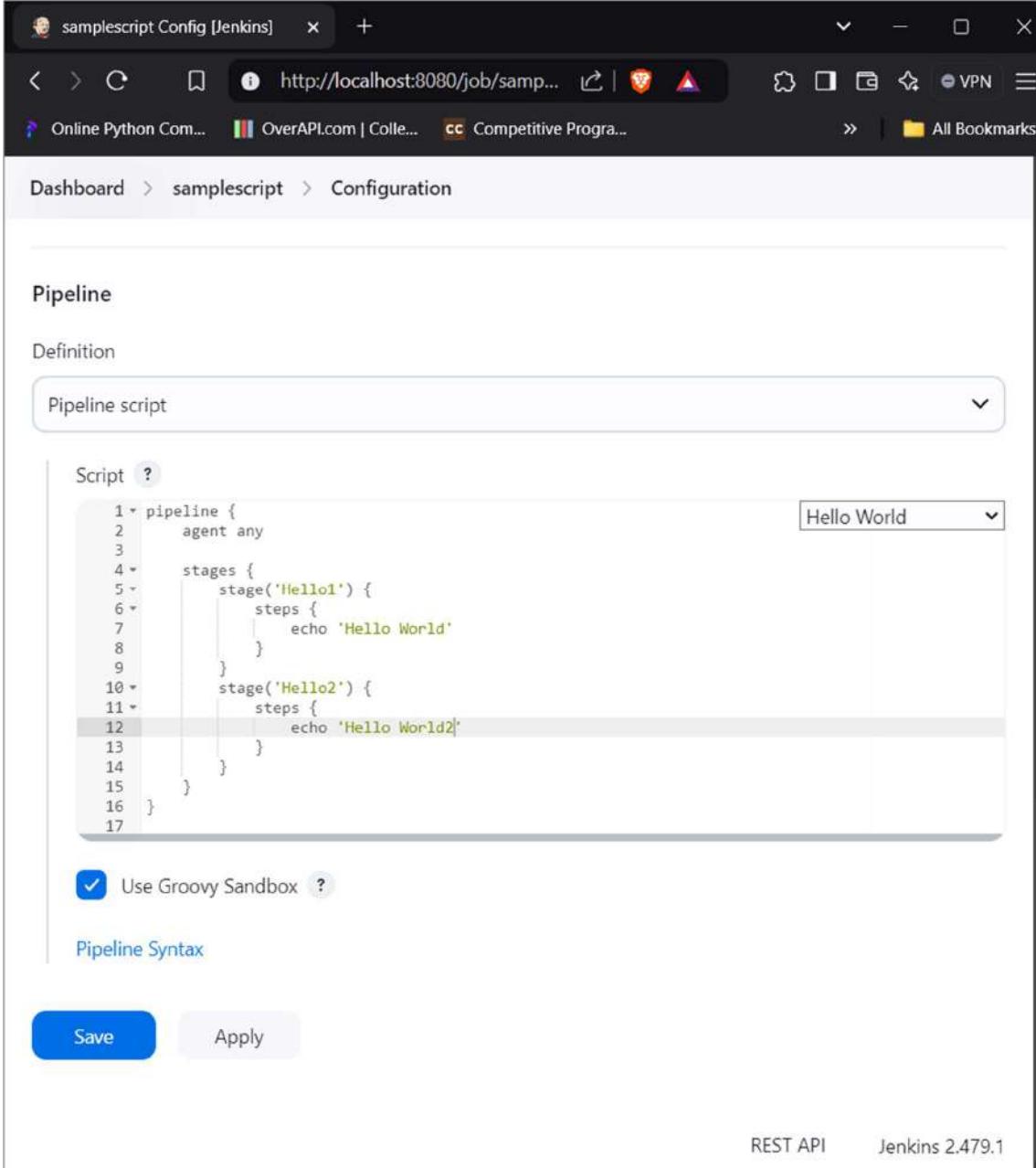
Configuring Maven

The screenshot shows the Jenkins configuration interface for adding a new Maven installation. The 'Maven installations' section is active, displaying fields for 'Name' (set to 'MAVEN_HOME'), 'MAVEN_HOME' (set to 'C:\Users\abdun\OneDrive\Desktop\apache-maven-3.9.9-bin\apache-maven-3.9.9'), and an unchecked 'Install automatically' checkbox. Below this is an 'Add Maven' button. At the bottom of the configuration page are 'Save' and 'Apply' buttons. In the top right corner, the text 'Jenkins 2.479.1' is visible.

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

The screenshot shows the Jenkins dashboard with a new item being created. The URL in the browser is 'http://localhost:8080/view/all/n...'. The 'New Item' dialog is open, showing the 'New Item' heading and a text input field containing 'samplescript'. Below it, a 'Select an item type' section lists several options: 'Freestyle project', 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder'. The 'Pipeline' option is highlighted with a light blue background. At the bottom of the dialog is an 'OK' button.

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



The screenshot shows the Jenkins Pipeline configuration page for a job named "samplescript". The "Definition" section is set to "Pipeline script". The script editor contains the following Groovy code:

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

A dropdown menu is open over the line "Hello World" in the code editor, showing options like "Copy", "Paste", "Delete", "Format", "Replace", "Edit", "Run", and "Run & Edit". Below the editor, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom of the configuration page, there are "Save" and "Apply" buttons, and a footer with links to "REST API" and "Jenkins 2.479.1".

Apply and save it

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

The screenshot shows a Jenkins job named "samplescript". The top navigation bar includes links for "Online Python Com...", "OverAPI.com | Colle...", and "Competitive Progra...". The main content area displays the "samplescript" job. A "Stage View" section is highlighted, showing two stages: "Hello1" and "Hello2". Stage "Hello1" took 163ms and stage "Hello2" took 111ms. Below the stage view, a "Permalinks" section lists four items, all corresponding to build #3. At the bottom, the URL "localhost:8080/job/samplescript/multi-pipeline-graph" is shown, along with links for "REST API" and "Jenkins 2.479.1".

samplescript [Jenkins]

http://localhost:8080/job/samp...

Online Python Com... OverAPI.com | Colle... Competitive Progra...

Dashboard > samplescript >

Filter

Today

#3 12:10 PM

samplescript Add description

Stage View

Hello1	Hello2
163ms	111ms
163ms	111ms

#3 Nov 24 12:10 No Changes

Average stage times:
(Average full run time: ~1s)

Permalinks

- Last build (#3), 2 min 40 sec ago
- Last stable build (#3), 2 min 40 sec ago
- Last successful build (#3), 2 min 40 sec ago
- Last completed build (#3), 2 min 40 sec ago

localhost:8080/job/samplescript/multi-pipeline-graph REST API Jenkins 2.479.1

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     }
6     stages {
7         stage('git repo & clean') {
8             steps {
9                 bat "rmdir /s /q javaMavenE"
10                bat "git clone https://github.com/iamabdurrasheed/javaMavenE.git"
11                bat "mvn clean -f javaMavenE"
12            }
13        }
14        stage('install') {
15            steps {
16                bat "mvn install -f javaMavenE"
17            }
}

```

try sample Pipeline... ▾

Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

Script ?

```

15     steps {
16         bat "mvn install -f javaMavenE"
17     }
18     stage('test') {
19         steps {
20             bat "mvn test -f javaMavenE"
21         }
22     }
23     stage('package') {
24         steps {
25             bat "mvn package -f javaMavenE"
26         }
27     }
28 }
29 }
30 }
31

```

try sample Pipeline... ▾

Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

Click on apply → save

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

Script:

```

pipeline {
    agent any
    tools {
        maven 'MAVEN_HOME'
        git 'GIT_HOME'
    }
    stages {
        stage('Git Repository & Clean') {
            steps {
                // Remove the existing directory if it exists
                bat """
                if exist javaMavenE (
                    rmdir /s /q javaMavenE
                )
                ...
                bat 'git clone https://github.com/iamabdurrasheed/javaMavenE.git'
                bat 'mvn clean -f javaMavenE'
            }
        }
        stage('Install') {
            steps {
                bat 'mvn install -f javaMavenE'
            }
        }
        stage('Test') {
            steps {
                bat 'mvn test -f javaMavenE'
            }
        }
        stage('Package') {
    
```

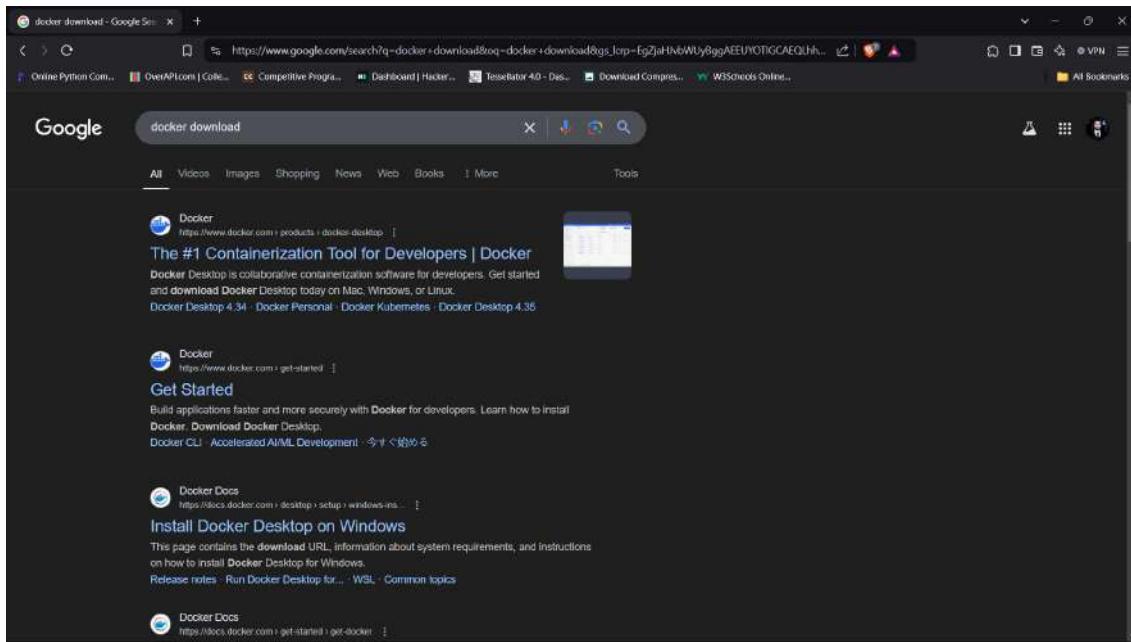
```
steps {  
    bat 'mvn package -f javaMavenE'  
}  
}  
}  
}  
}
```

The screenshot shows the Jenkins Pipeline script stage view. On the left, a sidebar lists pipeline-related actions: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main area is titled "Pipeline script" and "Stage View". It displays five stages: Declarative: Tool Install (287ms), Git Repository & Clean (5s), Install (11s), Test (7s), and Package (7s). Below these stages is a summary: "Average stage times: (Average full run time: -33s)". A tooltip for the "Tool Install" stage shows a timestamp from Nov 24 18:04 and a "1st Change" message. At the bottom, there's a "Permalinks" section with a "Builds" dropdown showing "Today" and "#4 2019PM". A navigation bar at the top includes "Dashboard", "Pipeline script", "Search (CTRL+K)", "Help", "Mohammed Abdur Rasheed", and "Logout".

Our stages are build

7 A. INSTALLATION OF DOCKER, MINIKUBE, ACCOUNT IN DOCKERHUB

Download the docker from the link as shown



Click on the following link to download the executable file



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

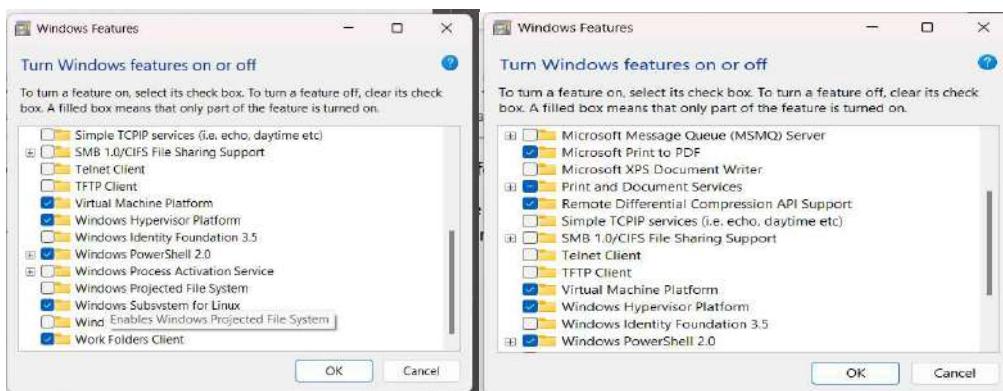


Make sure Virtualization is enabled in the system, we can check it in task manager as shown

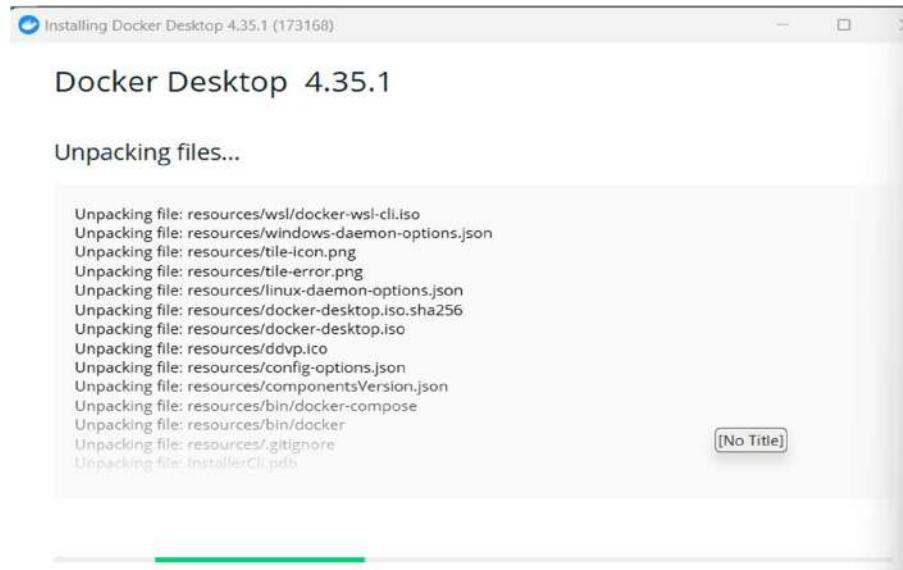


Click on turn windows features on or off in start

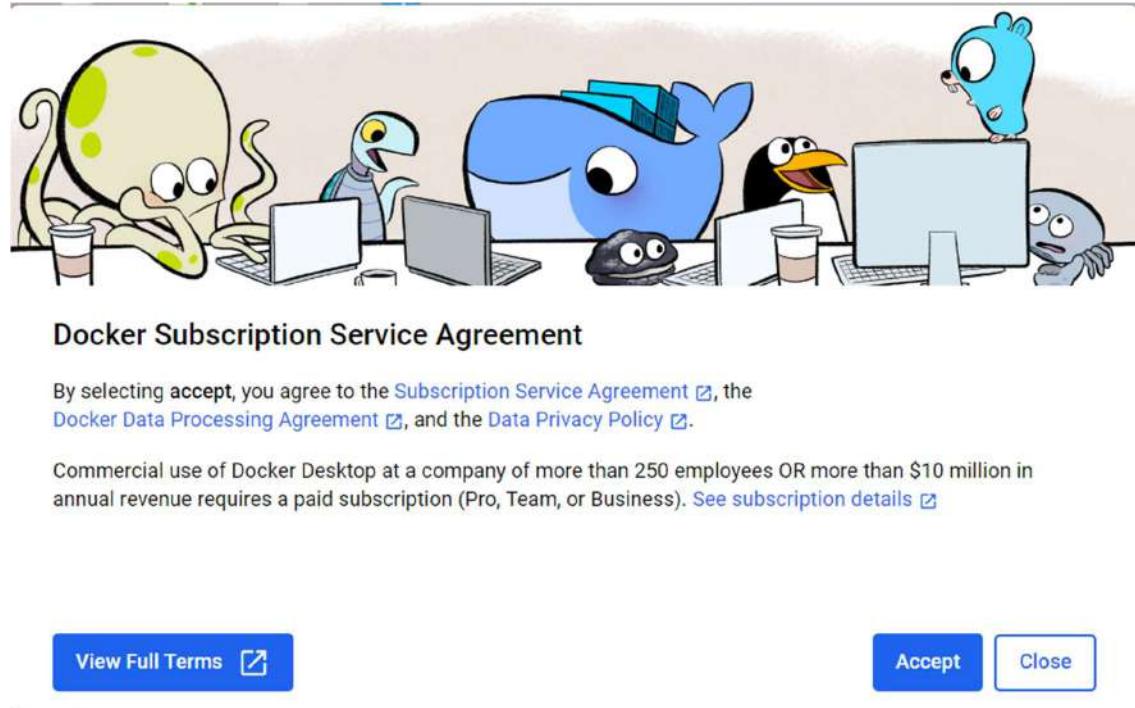
Make sure to check the Hyper-V, Virtual Machine Platform, Windows Hypervisor Platform and Windows subsystem for Linux checkboxes and click ok



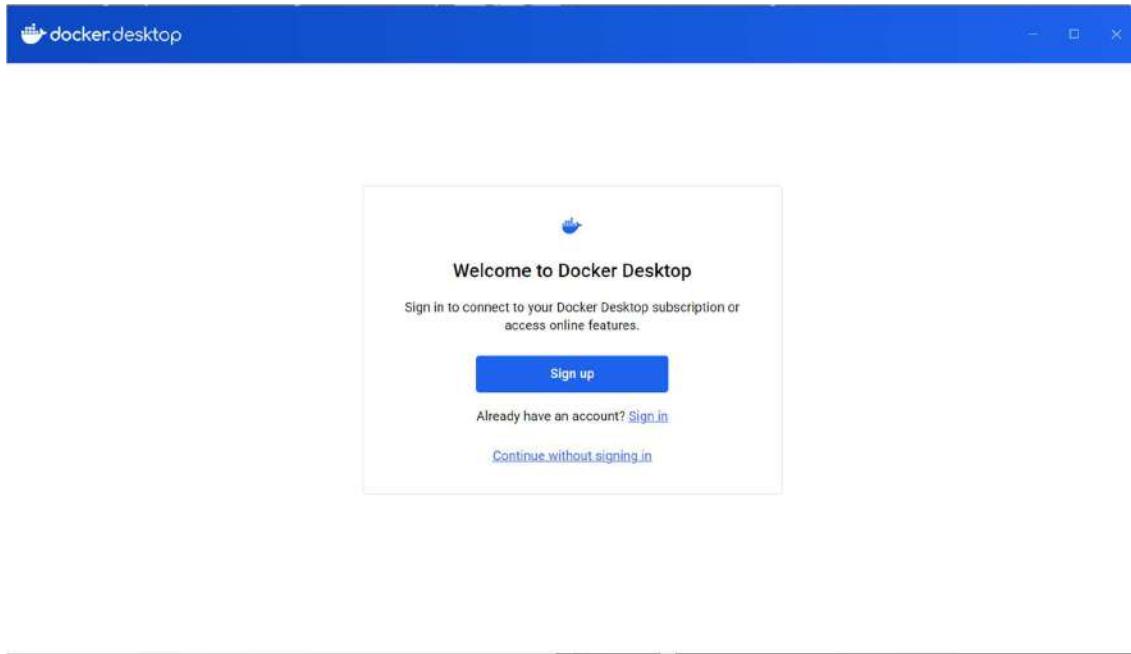
Start Docker installation ...



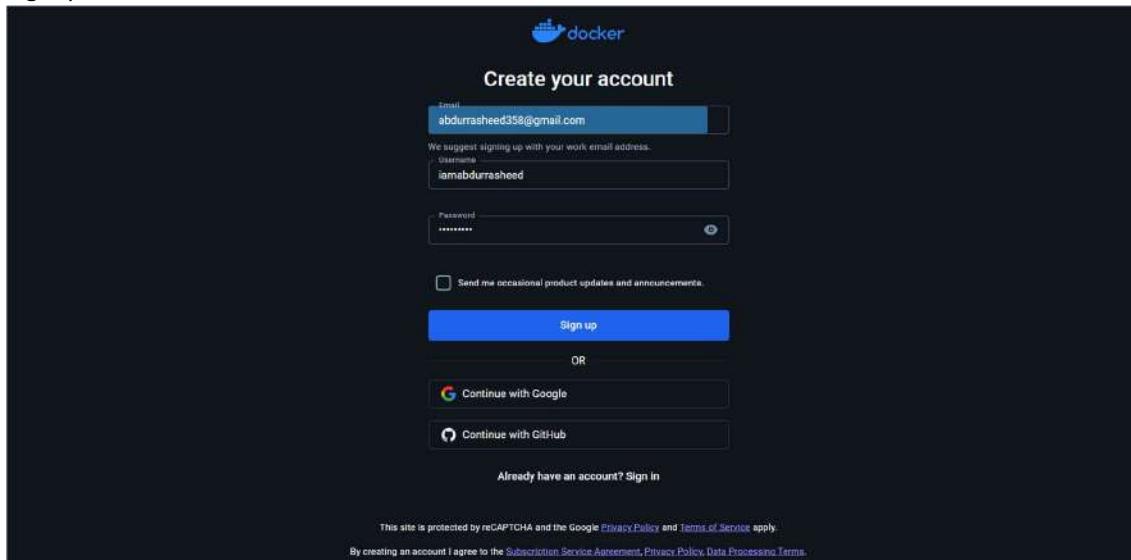
Accept the license -> click Accept



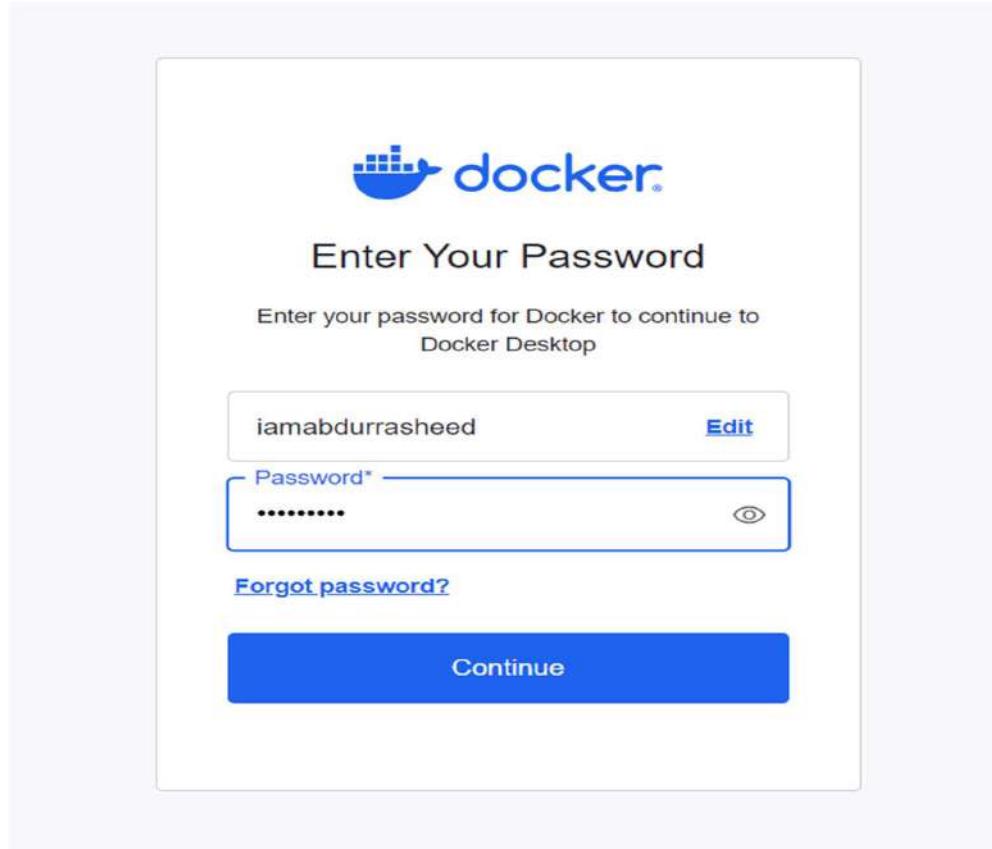
Create an Account in Docker..



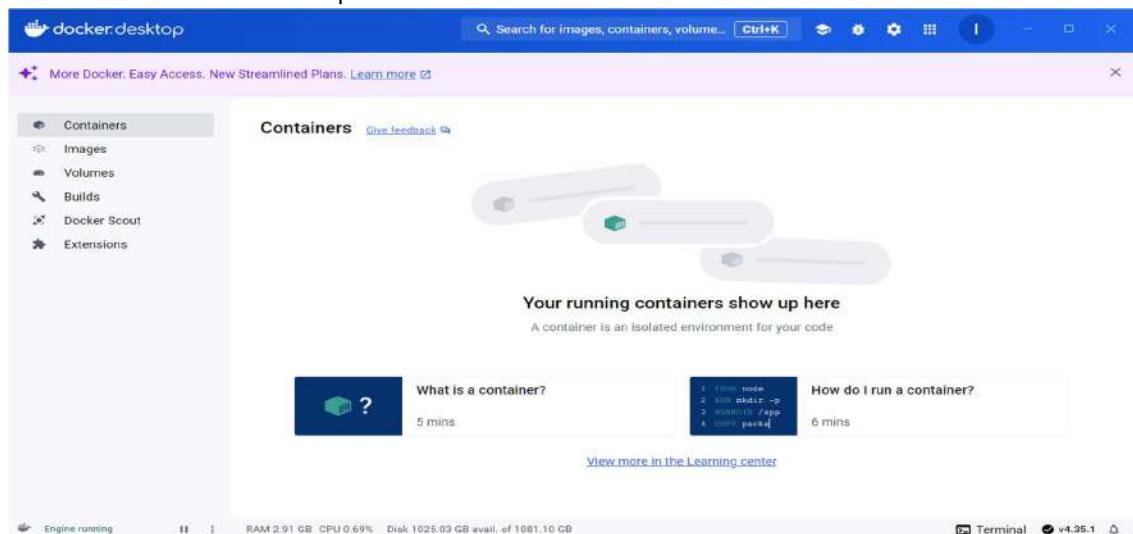
Signup ..



Now Sign in using your credentials

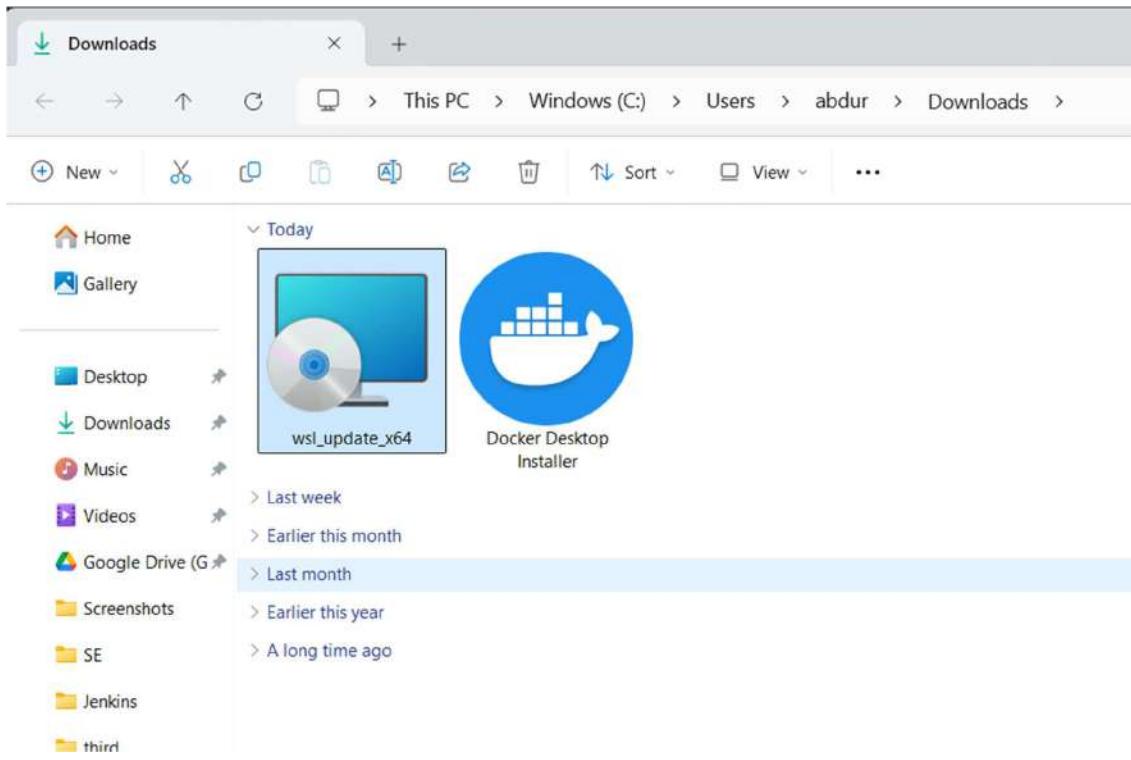


Redirected to Dockers Desktop

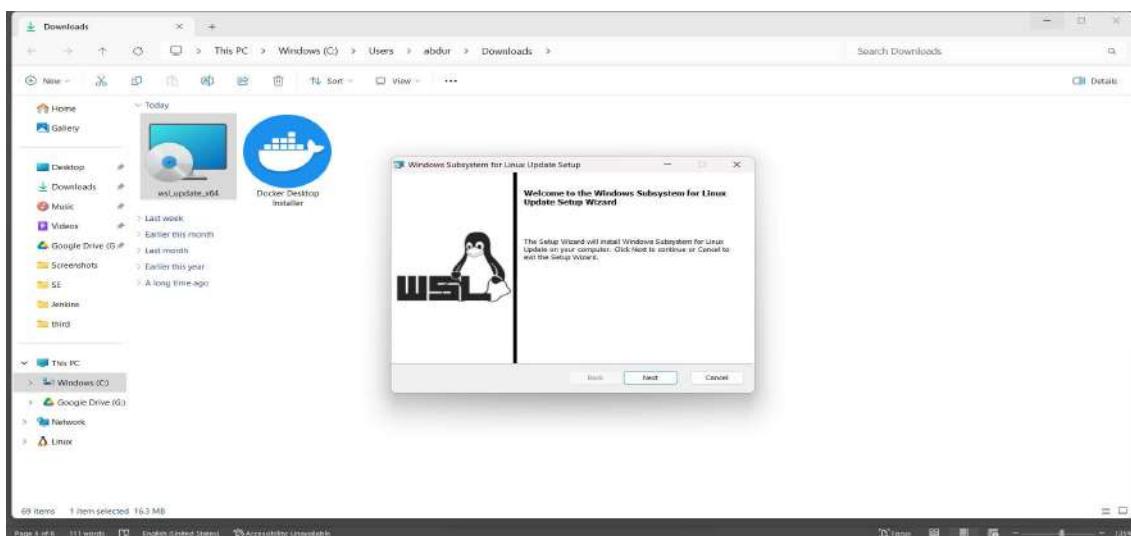


Download WSL2 from

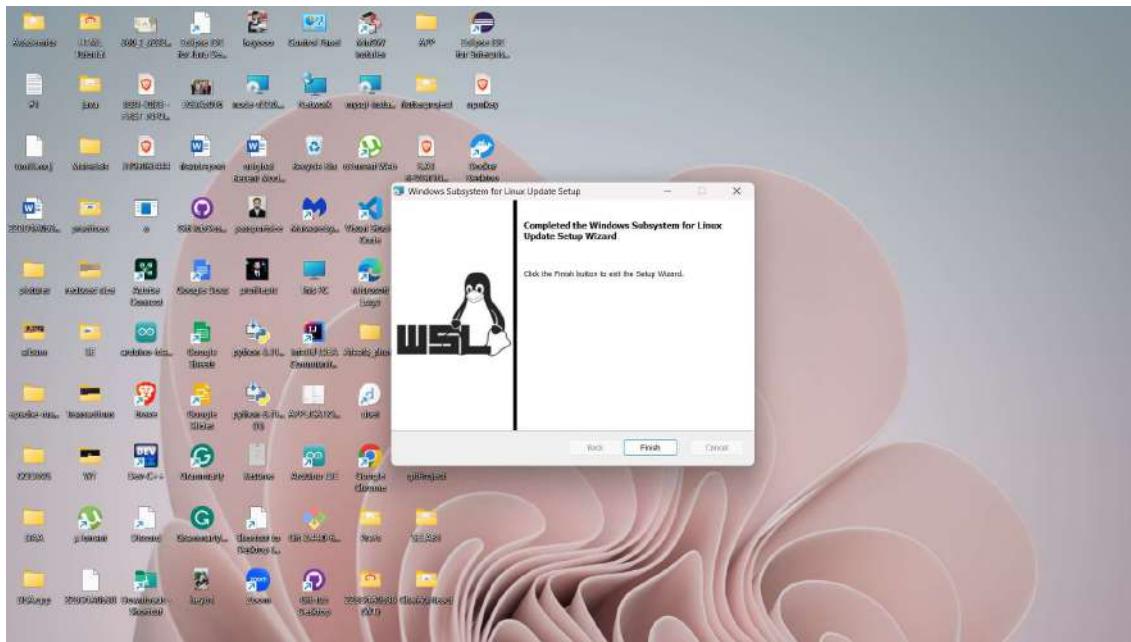
<https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package>



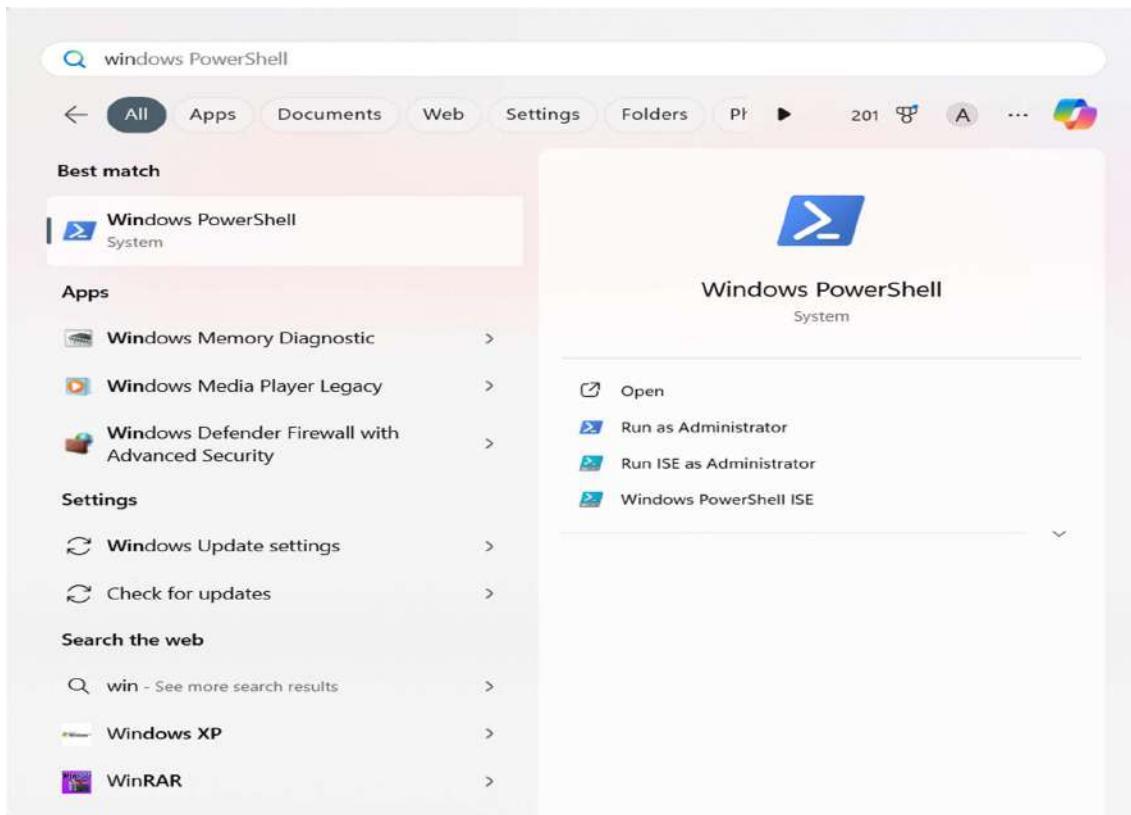
Open the setup wizard



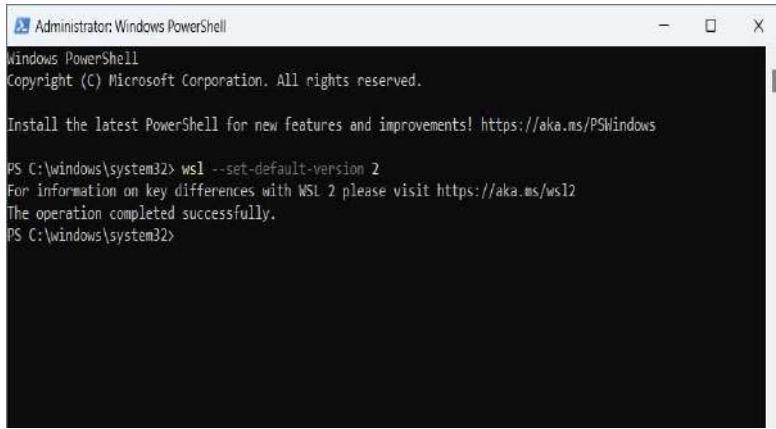
Click on Next → Allow Changes → Finish



Open Windows PowerShell as shown from Start→Run as administrator



Type the following command, to set WSL as default version



```
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>
```

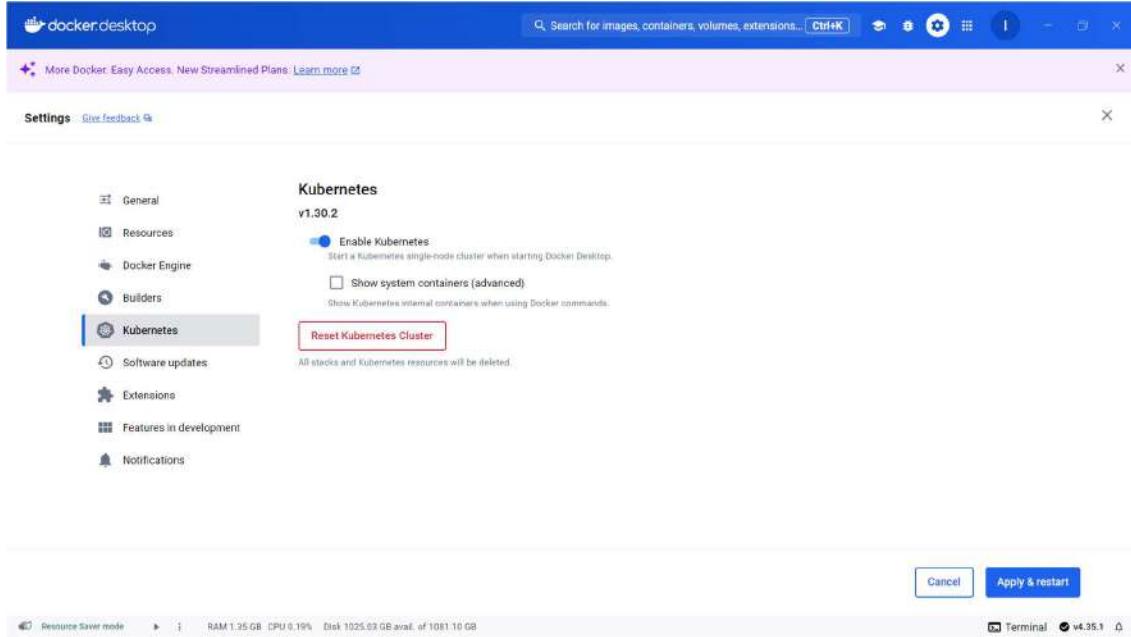
Now we can start the Docker

MINIKUBE INSTALLATION

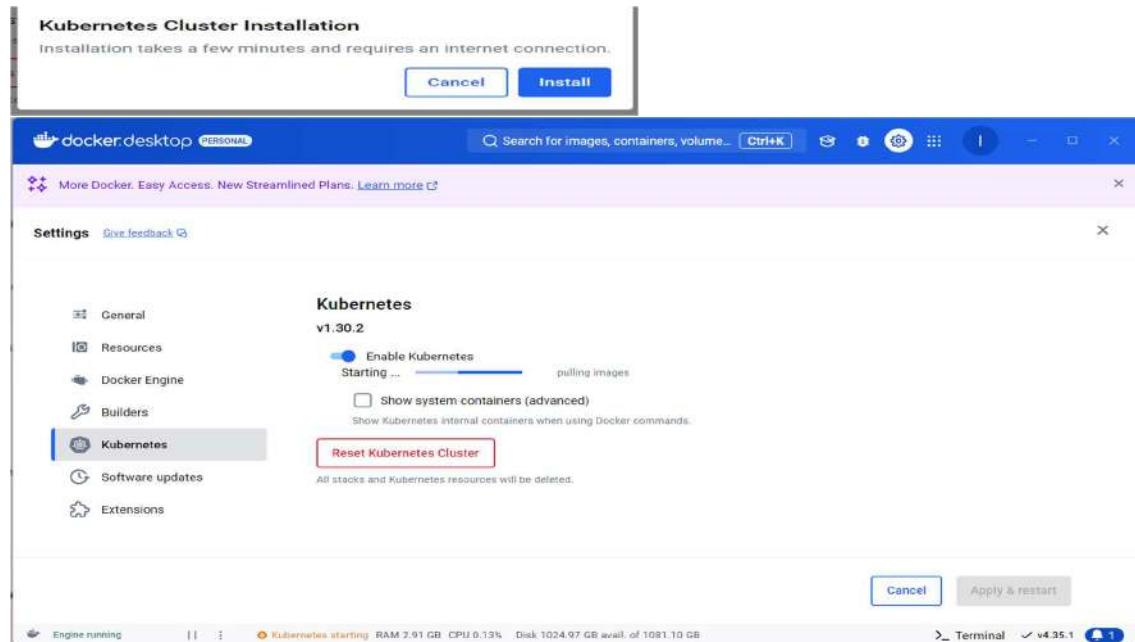
Minikube for Kubernetes using Docker Desktop

Install using Docker Desktop

Settings → Kubernetes → Check Enable Kubernetes → Apply

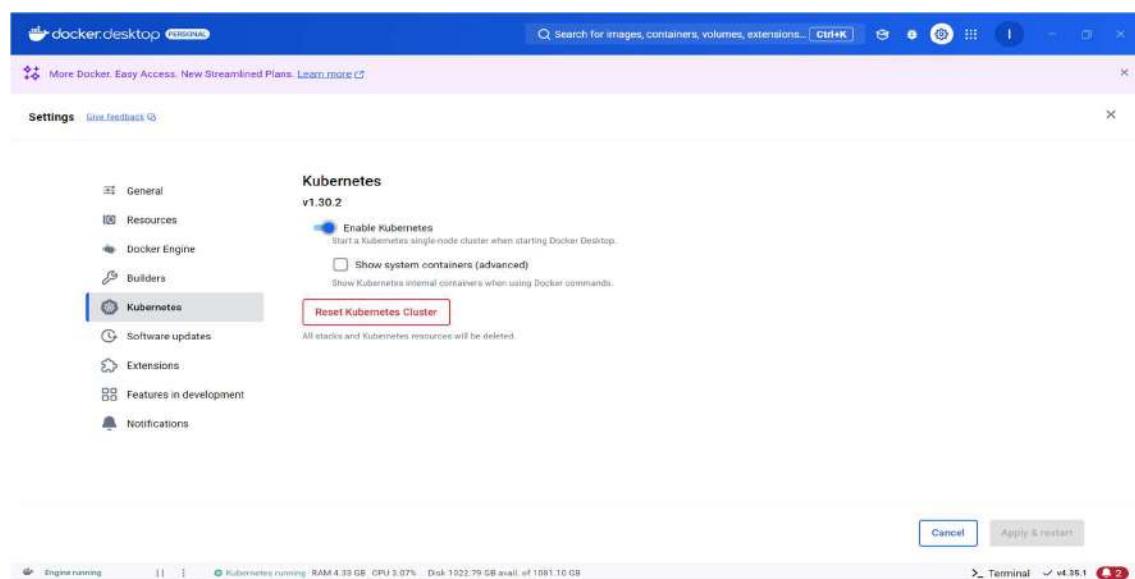


Pop Up click on Install



On the bottom Left

The green icons indicate that both services are running (docker and Kubernetes) successfully.



[Minikube](#) is a tool that allows you to run a single-node Kubernetes cluster inside a virtual machine, on your own computer.

<https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download>

Click on the buttons that describe your target platform. For other architectures, see the release page for a complete list of minikube binaries.

Operating system: Linux macOS Windows

Architecture: x86-64 ARM64

Release type: Stable Beta Alpha

Installer type: .exe download Windows Package Manager Chocolatey

To install the latest minikube **stable** release on **x86-64 Windows** using **Windows Package Manager**:

If the **Windows Package Manager** is installed, use the following command to install minikube:

```
winget install Kubernetes.minikube
```

[View page source](#) [Edit this page](#) [Create child page](#) [Create documentation issue](#) [What you'll need](#) [Take the next step](#)

Download the required minikube installer for the windows platform.

```
C:\Windows\system32\cmd.exe Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abdur\winget install Kubernetes.minikube
The 'msstore' source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires the current machine's 2-letter geographic region to be sent to the backend service to function properly (ex. "US").

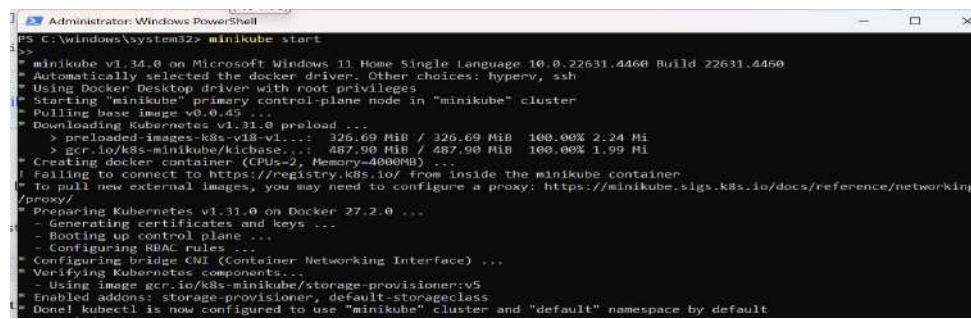
Do you agree to all the source agreements terms?
[Y] Yes [N] No: y
Found Kubernetes - Minikube - A Local Kubernetes Development Environment [Kubernetes.minikube] Version 1.34.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/kubernetes/minikube/releases/download/v1.34.0/minikube-installer.exe
Successfully verified installer hash
Starting package install...
Successfully installed
```

Run windows powershell as administrator

Command to set path

```
PS C:\Windows\system32> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
>>> if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
>>>   [Environment]::SetEnvironmentVariable('Path', ${'${0};C:\minikube'} -f $oldPath, [EnvironmentVariableTarget]::Machine)
>>> }
>>>
PS C:\Windows\system32>
```

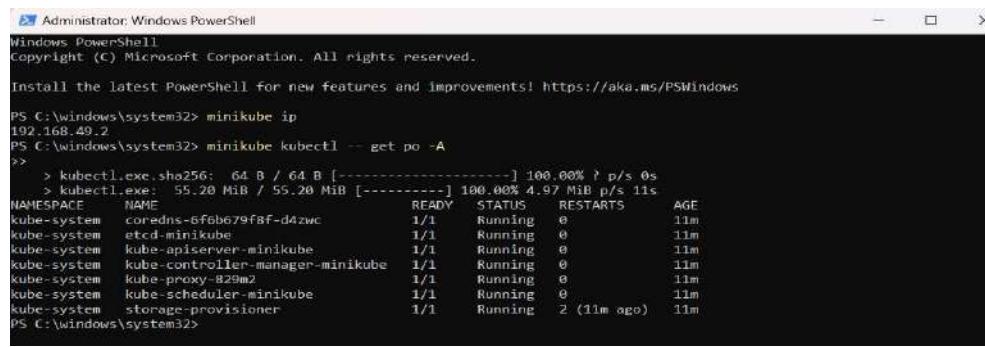
Start your cluster



```
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
* Automatically selected the docker driver. Other choices: hyperv, ssh
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.139.0 ...
* Downloading Kubernetes v1.31.0 preload ...
  * preloaded images-k8s-v18-v1...: 326.69 MiB / 326.69 MiB 100.00% 2.24 Mi
    > gcr.io/k8s-minikube/kicbase...
* Creating docker container (CPUs=2, Memory=4000MB) ...
! Failed to connect to https://registry.k8s.io/ from inside the minikube container
* 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 ...
* Generating certificates and keys ...
* Booting up control plane ...
* Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
* Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Interact with your cluster Using
minikube ip

kubectl get po -A to see the nodes



```
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

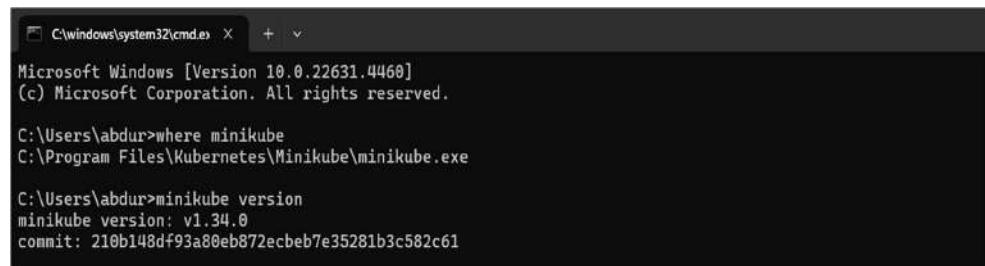
PS C:\windows\system32> minikube ip
192.168.49.2
PS C:\windows\system32> minikube kubectl -- get po -A
>>
  > kubelet.exe sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
  > kubelet.exe: 55.20 MiB / 55.20 MiB [-----] 100.00% 4.97 MiB p/s 11s
NAMESPACE     NAME           READY   STATUS    RESTARTS   AGE
kube-system   coredns-8f6b679f8f-d4zw
              1/1     Running   0          11m
kube-system   etcd-minikube
              1/1     Running   0          11m
kube-system   kube-apiserver-minikube
              1/1     Running   0          11m
kube-system   kube-controller-manager-minikube
              1/1     Running   0          11m
kube-system   kube-proxy-829m2
              1/1     Running   0          11m
kube-system   kube-scheduler-minikube
              1/1     Running   0          11m
kube-system   storage-provisioner
              1/1     Running   2 (11m ago) 11m
PS C:\windows\system32>
```

To check if path has been set

→ windows+r → type “cmd” → Enter → execute the following commands where minikube

:If it isn't set, you'll see an error message like: → INFO: Could not find files for the given pattern(s).
minikube version

:If you get a "command not found" or similar error, the path is not set.



```
C:\windows\system32\cmd.exe X + \v
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abdur>where minikube
C:\Program Files\Kubernetes\Minikube\minikube.exe

C:\Users\abdur>minikube version
minikube version: v1.34.0
commit: 210b148df93a80eb872ecbebe7e35281b3c582c61
```

7 B. DOCKER CLI COMMANDS

DOCKER COMMANDS

Check docker version using docker --version

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

Logging into docker

```
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 login -u iamabdurrasheed
Password:
Login Succeeded
PS C:\windows\system32>
```

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

```
Administrator: Windows PowerShell
PS C:\windows\system32> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Download complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
PS C:\windows\system32> docker run hello-world
```

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/
```

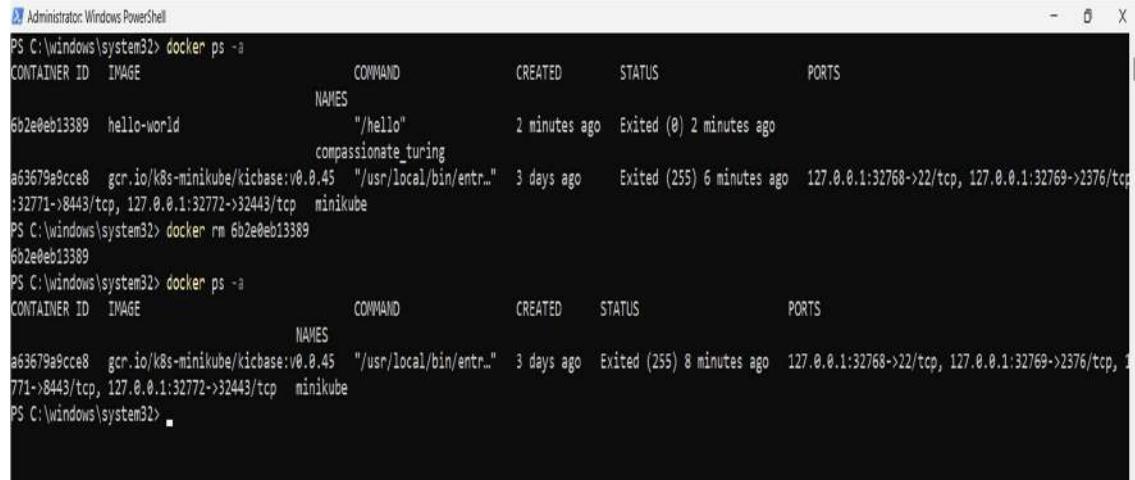
Lists all running Docker containers using docker ps

```
PS C:\windows\system32> docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
```

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

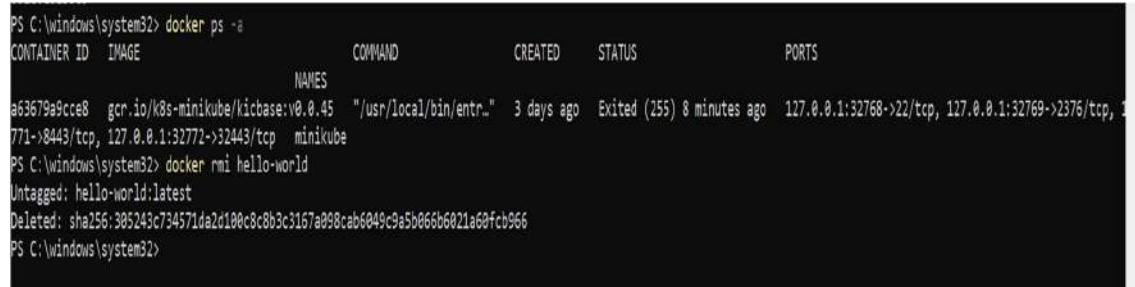
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker ps -a
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
6b2e0eb13389   hello-world   "/hello"      2 minutes ago  Exited (0) 2 minutes ago
                                         compassionate_turing
a63679a9cce8   gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr..."  3 days ago   Exited (255) 6 minutes ago  127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp
                                         ;32771->8443/tcp, 127.0.0.1:32772->32443/tcp  minikube
```

Removing a stopped Docker container using docker rm.



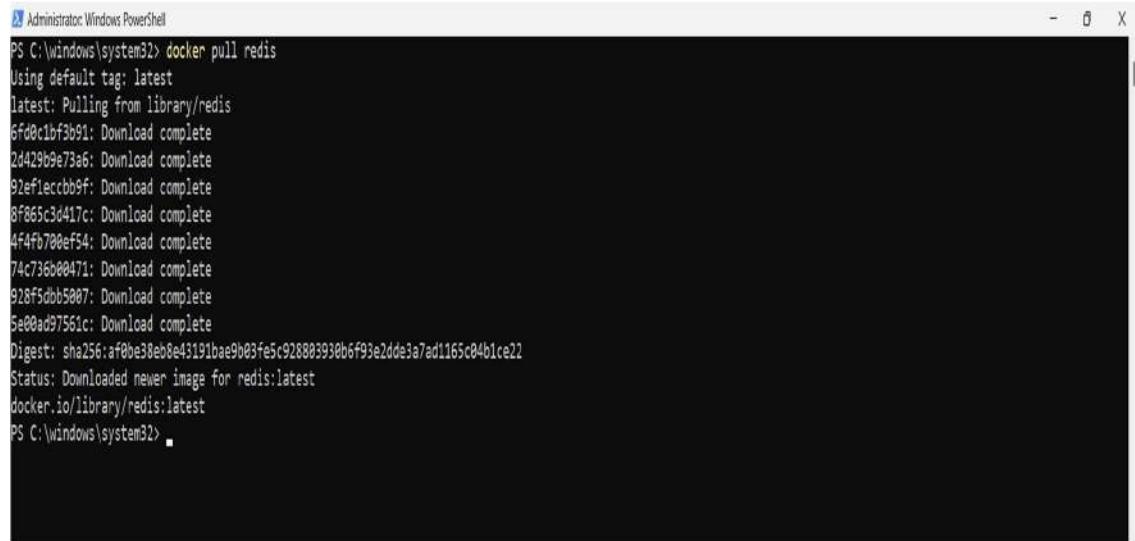
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
 NAMES
6b2e0eb13389 hello-world "/hello" 2 minutes ago Exited (0) 2 minutes ago
compassionate_turing
a63679a9cce8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 3 days ago Exited (255) 6 minutes ago 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp
:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\windows\system32> docker rm 6b2e0eb13389
6b2e0eb13389
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
 NAMES
a63679a9cce8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 3 days ago Exited (255) 8 minutes ago 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 1
771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\windows\system32>
```

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



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
 NAMES
a63679a9cce8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 3 days ago Exited (255) 8 minutes ago 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 1
771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\windows\system32> docker rmi hello-world
Untagged: hello-world:latest
Deleted: sha256:385243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fc9b966
PS C:\windows\system32>
```

Downloading a Docker image from docker hub using docker pull.



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
6f0c1bf3b91: Download complete
2d429b9e73a6: Download complete
92ef1eccbb9f: Download complete
8f865c3d417c: Download complete
4fafb70aeef54: Download complete
74c736b00471: Download complete
928f5db5907: Download complete
5e00ad97561c: Download complete
Digest: sha256:a90be38e08e43191bae9b03fe5c928803938b6f93e2dde3a7ad1165c04b1ce22
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.

```
PS C:\windows\system32> docker run --name newredis -d redis  
e07cb60b00fa07c8767a61ed64542a9f2e36bcd84aa7e9be27c3f082b147b3c6  
PS C:\windows\system32> .
```

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

```
Administrator: Windows PowerShell  
PS C:\windows\system32> docker exec -it newredis redis-cli  
127.0.0.1:6379> SET name "ABCDEFG"  
OK  
127.0.0.1:6379> GET name  
"ABCDEFG"  
127.0.0.1:6379> exit  
PS C:\windows\system32>
```

Stopping the newredis container that is running.

```
PS C:\windows\system32> docker exec -it newredis redis-cli  
127.0.0.1:6379> SET name "ABCDEFG"  
OK  
127.0.0.1:6379> GET name  
"ABCDEFG"  
127.0.0.1:6379> exit  
PS C:\windows\system32> docker stop newredis  
newredis  
PS C:\windows\system32> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
PS C:\windows\system32>
```

Starting the newredis container.

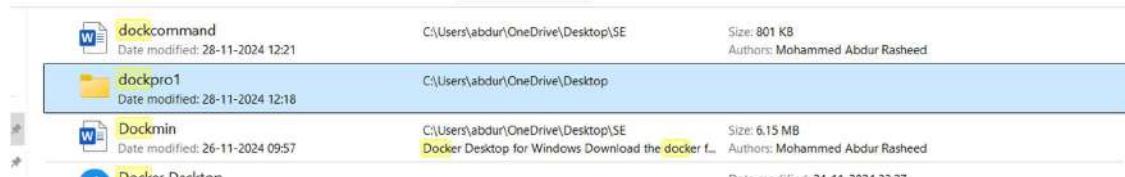
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker start newredis
newredis
PS C:\windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e07cb60b0bfa redis "docker-entrypoint.s..." 8 minutes ago Up 4 seconds 6379/tcp newredis
PS C:\windows\system32>
```

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

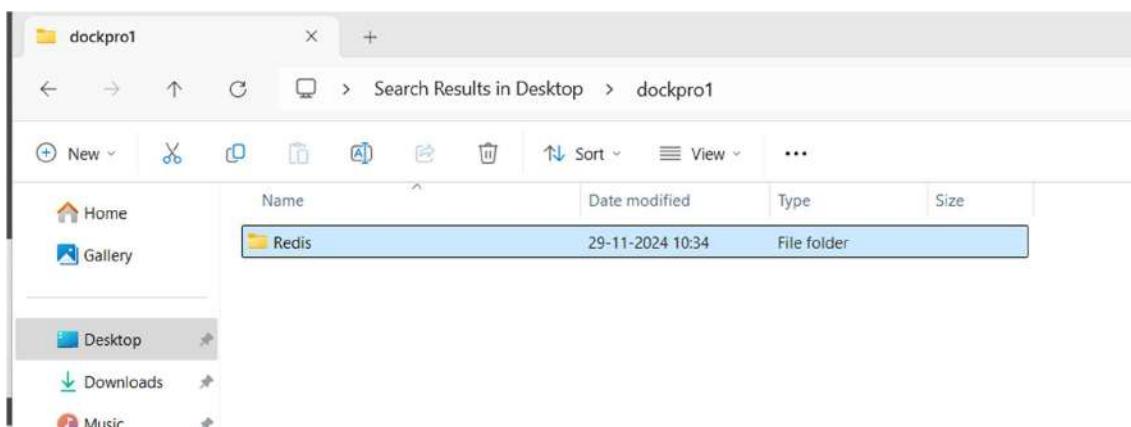
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e07cb60b0bfa redis "docker-entrypoint.s..." 8 minutes ago Up 4 seconds 6379/tcp newredis
PS C:\windows\system32> docker stop newredis
newredis
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
NAMES
e07cb60b0bfa redis "docker-entrypoint.s..." 12 minutes ago Exited (0) 12 seconds ago
newredis
a63679a9cce8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 3 days ago Exited (255) 27 minutes ago 12
.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
PS C:\windows\system32> docker rm e07cb60b0bfa
e07cb60b0bfa
PS C:\windows\system32> docker rmi redis
Untagged: redis:latest
Deleted: sha256:af0be38eb8e43191bae9b03fe5c928803930b6f93e2dde3a7ad1165c04b1ce22
```

CREATING A CUSTOM DOCKER FILE

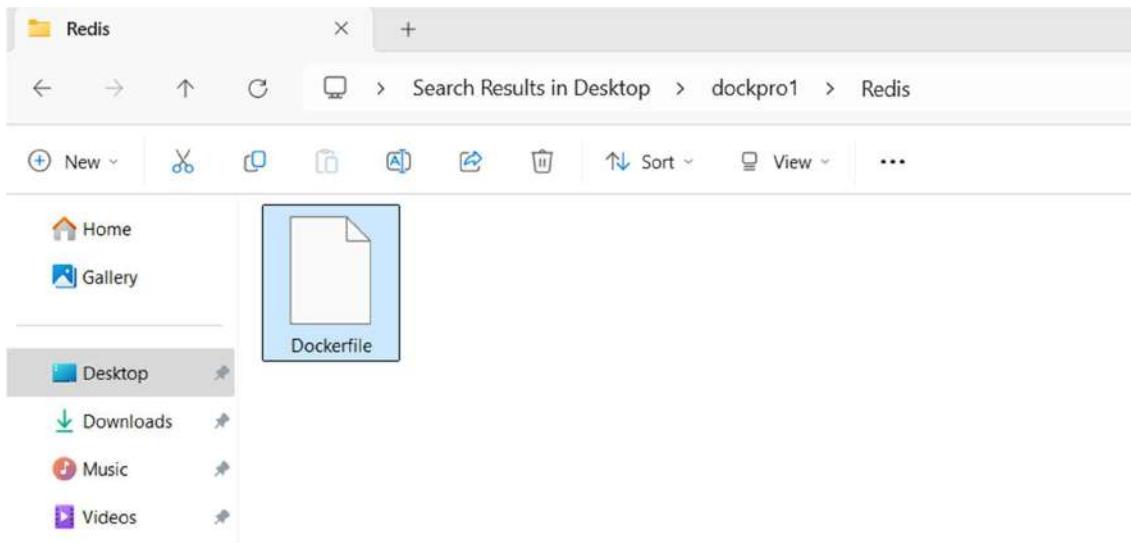
Creating a new folder named dockpro1



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.

```
FROM redis:latest
CMD ["redis-server"]
```

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

```
PS C:\windows\system32> cd C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

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

```
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker build -t redisnew .
>>
[+] Building 21.1s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> transfering dockerfile: 76B
=> [internal] load metadata for docker.io/library/redis:latest
=> [auth] library/redis:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> transfering context: 28
=> [1/1] FROM docker.io/library/redis:latest@sha256:aef0be38eb8e43191bae9b03fe5c928803930b6f93e2dde3a7ad1165c04b1ce22
=> resolve docker.io/library/redis@sha256:aef0be38eb8e43191bae9b03fe5c928803930b6f93e2dde3a7ad1165c04b1ce22
=> sha256:92ef1ecbb9f0c2337e86a92a94b0f2cb943e57cf07cb4f6543cd987d0bf25e1 1.10kB / 1.10kB
=> sha256:4f4fb700ef54a61cfab2571ae0dd9a0dc1e0ccb55774846d75e68dc38e8acc1 32B / 32B
=> sha256:9f865c3d417c3b5774414c517cd9c664e899ae51f7e5c837fdc63f99e63b9cb 1.4MB / 1.4MB
=> sha256:74c736000471cb382d7e0e535b8eeb79a325fb394530cc0de4da7758d8bfef0a 15.33MB / 15.33MB
=> sha256:92af5dbb5007ea861e1dc501806c842cf72546ff88d55:9812758f9df3b93f 968 / 968
=> sha256:2d429b9e73a6cf9805bb85105c8118b3@0a1b2deedea3ea9587055ffcb80b45 29.13MB / 29.13MB
=> sha256:f6f00c1bf3b91f3d5955127ae04e988a24bffa39cafa45d2677f6159e2e68b7fe 572B / 572B
=> sha256:5e00ad97561c34a8330c0eb8a#76b04a6e84610f8d9cd5822c103ac247b1088 873B / 873B
=> exporting to image
=> exporting layers
=> => exporting manifest sha256:e618dc41f5633a6babae6c98e970e1a70d15116e551cbd2aadf39a5ac20e590e
=> => exporting config sha256:c7deefbfb0f05029a4d4f3773fe0a798b0b4bdas8f6cc02310599eb801658cd4e1
=> => exporting attestation manifest sha256:699ace0488bddc13c7d4429340a55595f236fc804843481f1218494e23dc6379a
=> => exporting manifest list sha256:a5b169adec4a010a6fd02c42f17e660327ec63bc27fcbb3c1e71a417ed1dec09a
=> => naming to docker.io/library/redisnew:latest
=> => unpacking to docker.io/library/redisnew:latest
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

Run the command docker images to view all the images available

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
redisnew	latest	a5b169adec4a	7 weeks ago	174MB
gcr.io/k8s-minikube/kicbase	<none>	81df28859520	2 months ago	1.81GB
gcr.io/k8s-minikube/kicbase	v0.0.45	e7c9bc3bc515	2 months ago	1.81GB
registry.k8s.io/coredns/coredns	v1.11.3	9caabbf6238b	4 months ago	85.1MB
docker/desktop-kubernetes	kubernetes-v1.30.2-cni-v1.4.0-critools-v1.29.0-cri-dockerd-v0.3.11-1-debian	c86a2c396272	4 months ago	625MB
registry.k8s.io/kube-apiserver	v1.30.2	340ab4a1d66a	5 months ago	153MB
registry.k8s.io/kube-controller-manager	v1.30.2	4c412bc1fc58	5 months ago	146MB
registry.k8s.io/kube-scheduler	v1.30.2	0ed75a333784	5 months ago	84.6MB
registry.k8s.io/kube-proxy	v1.30.2	8a44c6e094af	5 months ago	118MB
registry.k8s.io/etcld	3.5.12-0	44a8e24dc1bba	9 months ago	211MB
registry.k8s.io/coredns/coredns	v1.11.1	1eeb4c7316ba	15 months ago	82MB
docker/desktop-vpnkit-controller	dc331cb22850be0cd97c84a9cfecaf44a1af6e	7ecf567ea070	18 months ago	47MB
registry.k8s.io/pause	3.9	7031c1b28338	2 years ago	1.07MB
docker/desktop-storage-provisioner	v2.0	115d77e6e2	3 years ago	59.2MB

Running the new latest redis image that we created.

```

PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker run --name redislatest -d redisnew
f4d9149baa4a597c2fc9c792f16876ee01874e54f432a9eb0ee67aa67c5e8ce3
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>

```

Viewing the docker container that are running.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f4d9149baa4a	redisnew	"docker-entrypoint.s..."	4 minutes ago	Up 4 minutes	6379/tcp	redislatest

Stopping the docker container od redislatest.

```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker run -name redislatest -d redisnew
f4d9149baa4a597c2fc9c792f16876ee01874e54f432a9eb0ee67aa67c5e8ce3
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f4d9149baa4a redisnew "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 6379/tcp redislatest
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker stop f4d9149baa4a
f4d9149baa4a
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\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

```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f4d9149baa4a redisnew "docker-entrypoint.s..." 9 minutes ago Exited (0) 2 minutes ago
a63679a9cce8 gcr.io/k8s-minikube/kicbase:v0.0.45 "/usr/local/bin/entr..." 4 days ago Exited (255) 24 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\abdur\OneDrive\Desktop\dockpro1\Redis> docker commit f4d9149baa4a iamabdurrasheed/redisnew
>>
>>
sha256:951f8b364af11be08c34a6b6a7d4bec8cccd31e47739730c3b3484221e16627
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

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

Name	Tag	Image ID	Created	Size	Actions
registry.k8s.io/pause	3.9	7031c1b28338	2 years ago	1.06 MB	⋮ trash
docker/desktop-storage	v2.0	115d77ef6e2	4 years ago	59.16 MB	⋮ trash
redisnew	latest	a5b169adec4a	2 months ago	173.68 MB	⋮ trash
iamabdurrasheed	latest	0976a4de7910	2 minutes ago	173.68 MB	⋮ trash
iamabdurrasheed/redisr	latest	951f8b364af1	1 minute ago	173.68 MB	⋮ trash

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
8f865c3d417c: Pushed
92bf5dbb5007: Pushed
4f4fb700ef54: Pushed
2d429b9e73a6: Pushed
92ef1eccbb9f: Pushed
6fd0c1bf3b91: Pushed
5e00ad97561c: Pushed
latest: digest: sha256:81df38b11bc2223b89aedf3fd238fa7f206e067a2eb0e95c0b146d6160efa5f size: 2038
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

Image after it is pushed into the docker hub.

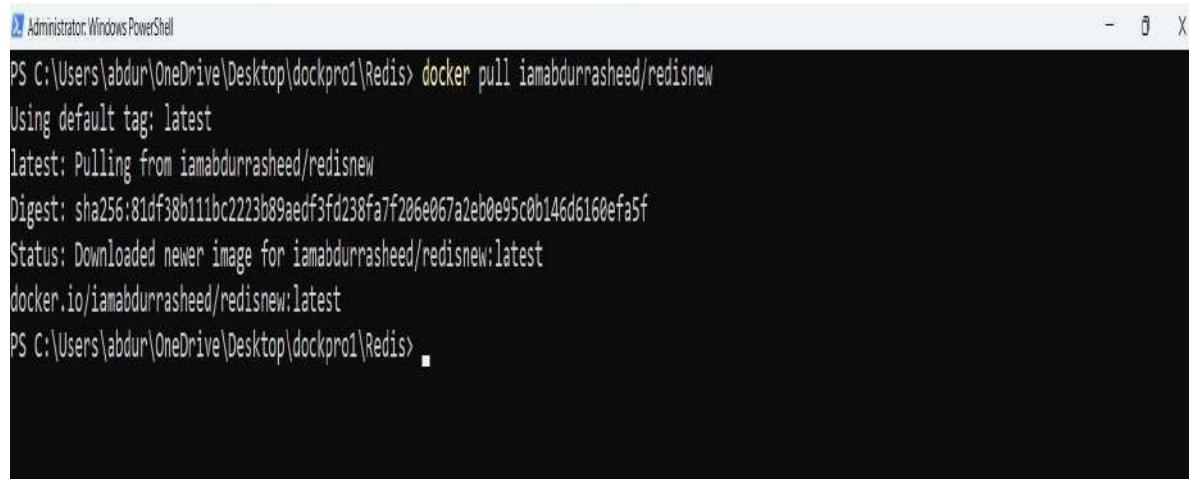
The screenshot shows the Docker Hub interface. At the top, there are navigation tabs: Explore, **Repositories**, Organizations, Usage, and a search bar labeled "Search Docker Hub". Below the search bar, there's a dropdown menu set to "iamabdurrasheed" and a "Create a repository" button. The main area displays a table of repositories. The first row in the table is "iamabdurrasheed/redisnew", which was last pushed 5 minutes ago, is an IMAGE, has Public visibility, and is inactive. To the right of the table, there's a decorative graphic featuring a red circle with a white icon, a blue hexagon with a lock icon, and a green triangle with a gear icon.

Name	Last Pushed	Contains	Visibility	Scout
iamabdurrasheed/redisnew	5 minutes ago	IMAGE	Public	Inactive

Now removing the images and container locally.

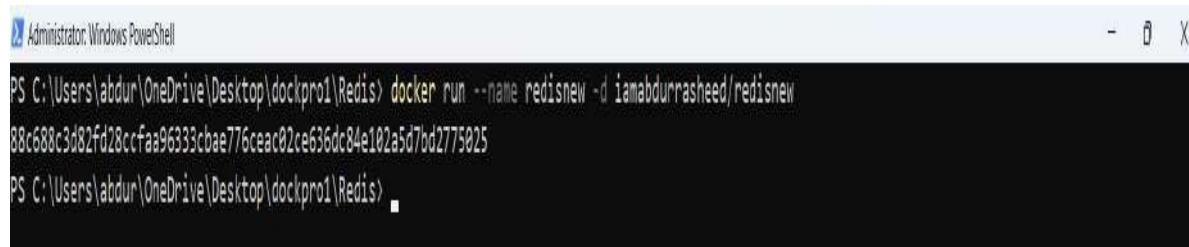
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              PORTS
f4d9149baa4a   redisnew      "docker-entrypoint.s..."  25 minutes ago   Exited (0) 19 minutes ago
                                                               NAMES
                                                               redislatest
a63679a9cce8   gcr.io/k8s-minikube/kicbase:v0.0.45  "/usr/local/bin/entr..."  4 days ago       Exited (255) 25 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\abdur\OneDrive\Desktop\dockpro1\Redis> docker rm f4d9149baa4a
f4d9149baa4a
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker rmi iamabdurrasheed/redisnew
Untagged: iamabdurrasheed/redisnew:latest
Deleted: sha256:81df38b11bc2223b89aedf3fd238fa7f206e067a2eb0e95c0b146d6160efa5f
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

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



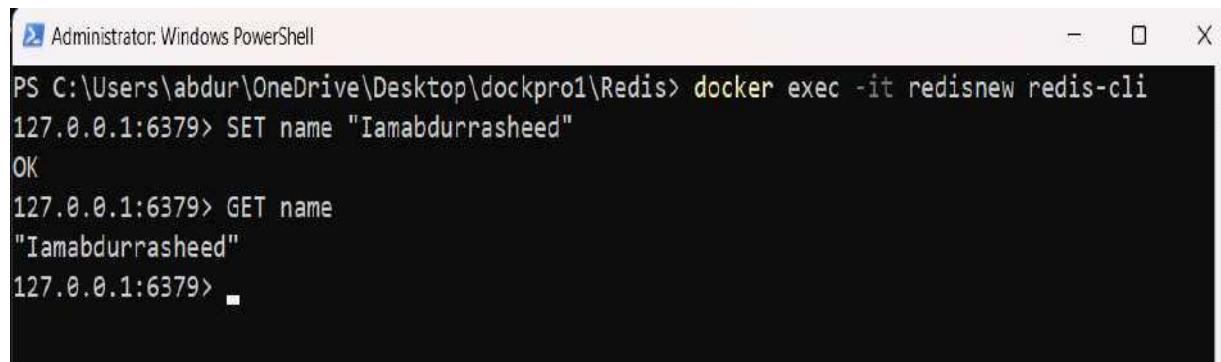
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker pull iamabdurrasheed/redisnew
Using default tag: latest
latest: Pulling from iamabdurrasheed/redisnew
Digest: sha256:81df38b111bc2223b89aedf3fd238fa7f206e067a2eb0e95c0b146d6160efa5f
Status: Downloaded newer image for iamabdurrasheed/redisnew:latest
docker.io/iamabdurrasheed/redisnew:latest
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

Running the new redis file just we just pulled.



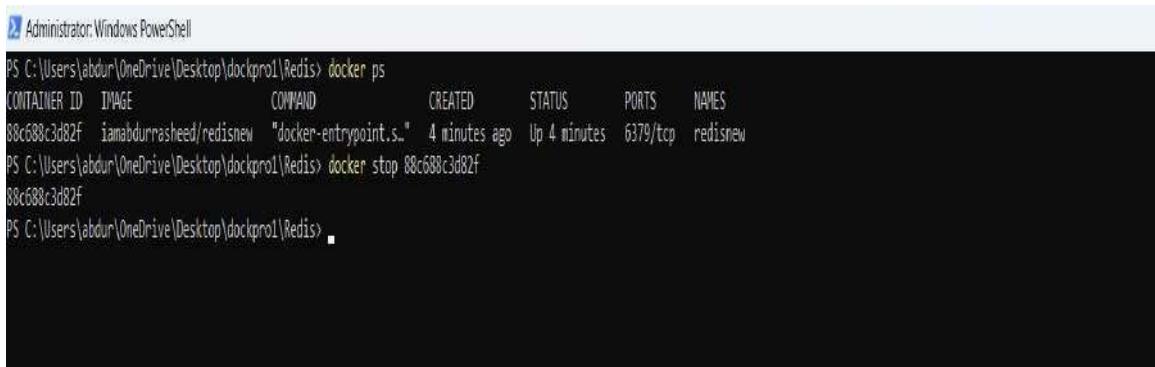
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker run --name redisnew -d iamabdurrasheed/redisnew
88c688c3d82fd28ccfaa96333cbae776ceac02ce636dc84e102a5d7bd2775025
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

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



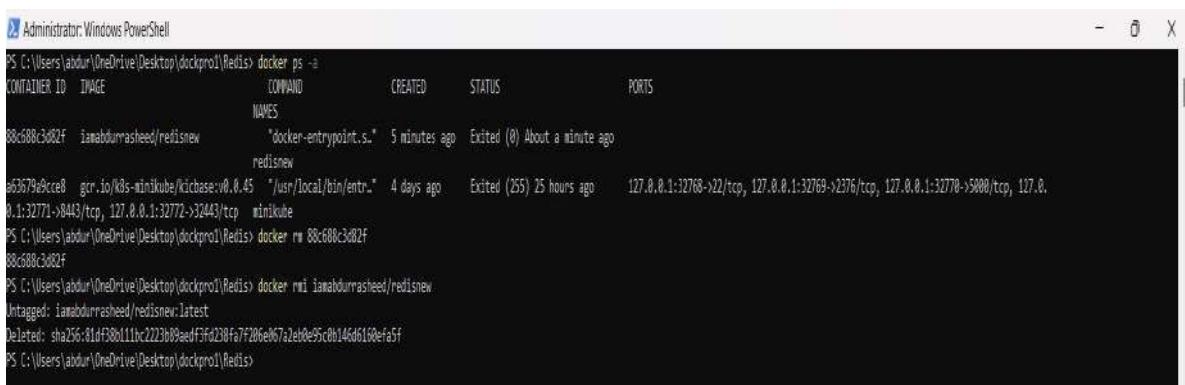
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker exec -it redisnew redis-cli
127.0.0.1:6379> SET name "Iamabdurrasheed"
OK
127.0.0.1:6379> GET name
"Iamabdurrasheed"
127.0.0.1:6379>
```

Stopping the container that was running.



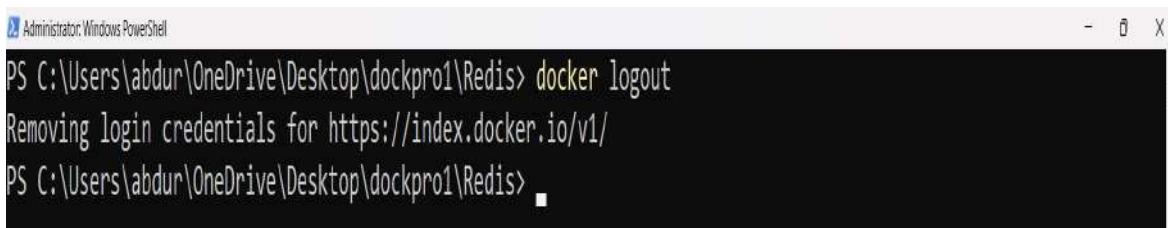
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
88c688c3d82f iamabdurrasheed/redisnew "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 6379/tcp redisnew
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker stop 88c688c3d82f
88c688c3d82f
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

Deleting the container and images after it is stopped.



```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
88c688c3d82f iamabdurrasheed/redisnew "docker-entrypoint.s..." 5 minutes ago Exited (0) About a minute ago
redisnew
a63679e9cce8 gcr.io/k8s-minikube/kicbase:v0.8.45 "/usr/local/bin/entry" 4 days ago Exited (255) 25 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\abdur\OneDrive\Desktop\dockpro1\Redis> docker rm 88c688c3d82f
88c688c3d82f
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker rmi iamabdurrasheed/redisnew
Untagged: iamabdurrasheed/redisnew:latest
Deleted: sha256:81df38d11bc2233b09addf3fd230f7f7206e06732eb0e95cb146d6168efa5f
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

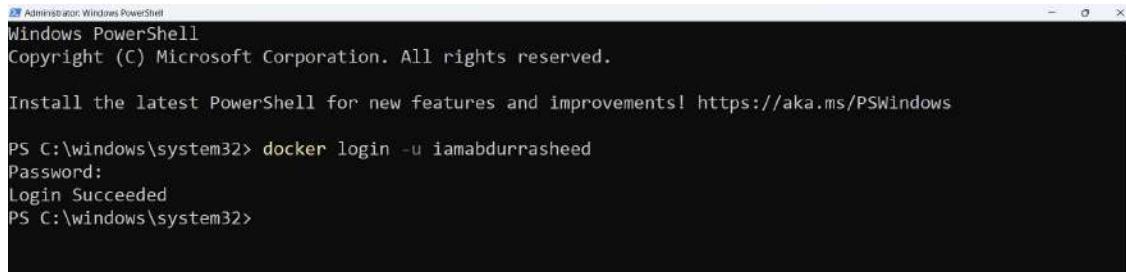
Logging out from the docker.



```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis> docker logout
Removing login credentials for https://index.docker.io/v1/
PS C:\Users\abdur\OneDrive\Desktop\dockpro1\Redis>
```

7 C. MODIFY AND PUSH DOCKER IMAGE

1. Login to the docker using docker login -u username. In administration mode

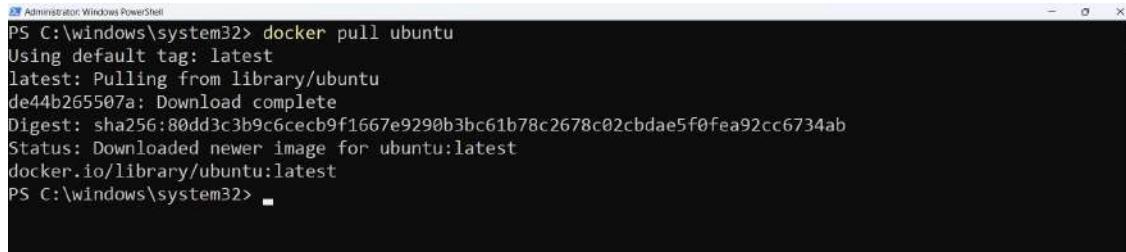


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

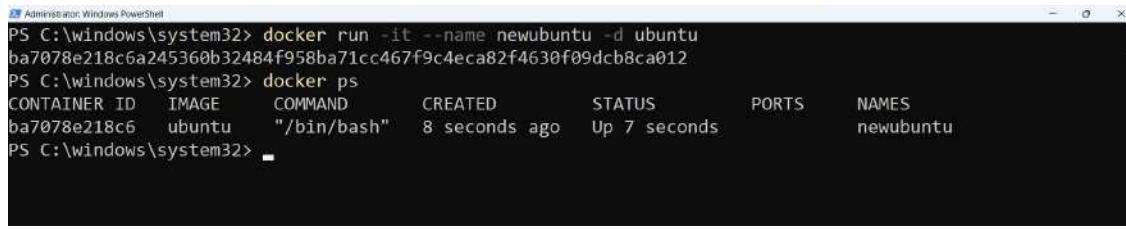
PS C:\windows\system32> docker login -u iamabdurrasheed
Password:
Login Succeeded
PS C:\windows\system32>
```

2. Let us first pull the ubuntu image . Pulling the latest ubuntu image.



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
de44b265507a: Download complete
Digest: sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\windows\system32>
```

3. Running the ubuntu image in a container that we just pulled.



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker run -it --name newubuntu -d ubuntu
ba7078e218c6a245360b32484f958ba71cc467f9c4eca82f4630f09dc8ca012
PS C:\windows\system32> docker ps
CONTAINER ID   IMAGE     COMMAND      CREATED       STATUS        PORTS     NAMES
ba7078e218c6   ubuntu    "/bin/bash"  8 seconds ago  Up 7 seconds          newubuntu
PS C:\windows\system32>
```

4. Now let us enter into ubuntu interactive operation of git.

```
PS C:\windows\system32> docker ps
CONTAINER ID   IMAGE     COMMAND      CREATED      STATUS       PORTS     NAMES
ba7078e218c6   ubuntu    "/bin/bash"  3 minutes ago Up 3 minutes   newubuntu
PS C:\windows\system32> docker exec -it ba7078e218c6 bash
root@ba7078e218c6:/#
```

5. Updating the apt using the command “apt update”.

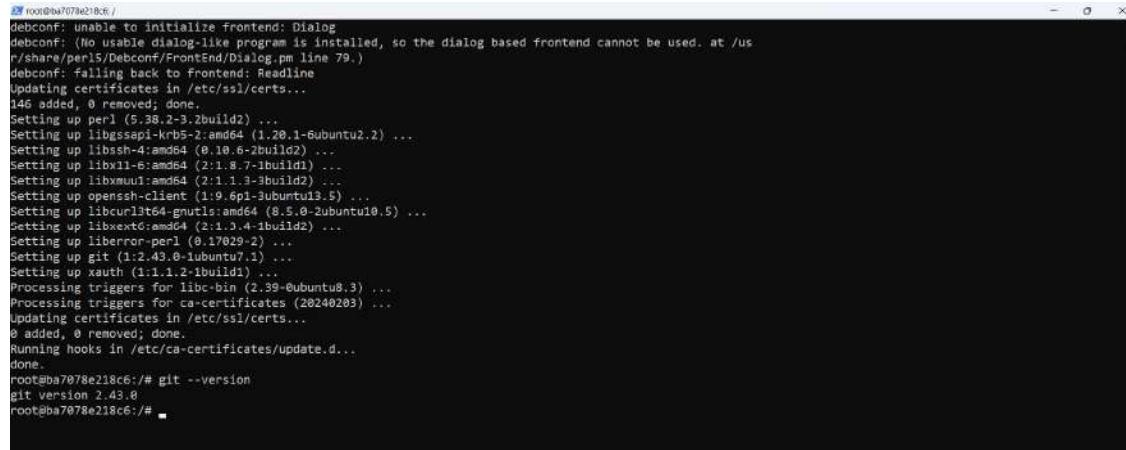
```
root@ba7078e218c6:/# git --version
git: command not found
root@ba7078e218c6:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [627 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [15.3 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [726 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [607 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1809 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [19.7 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [938 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [628 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [858 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [11.9 kB]
Fetched 26.6 MB in 45s (593 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ba7078e218c6:/#
```

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

```
root@ba7078e218c6:/# 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 libbrotli0 libbsd0 libcbor0.10 libcurl3t64-gnutls
  libedit2 liberror-perl libexpat1 libfdio2-1 libgdbm-compat4t64 libgdbm6t64 libgsapi-krb5-2
  libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 libnghttp2-14
  libperl5.38t64 libpsl5t64 librmp1 libssasl2-2 libssasl2-modules libssasl2-modules-db libssh-4 libx11-6
  libx11-data libxa6 libxcb1 libxdmcp6 libxext6 libxmuu1 netbase openssh-client openssl patch perl
  perl-modules-5.38 publicsuffix xauth
Suggested packages:
  liblocale-gettext-perl cron quota cryptfs-utils gettext-base git-daemon-run | git-daemon-sysvinit
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn gdm3-110 krb5-doc krb5-user
  libssasl2-modules-gssapi-mit | libssasl2-modules-gssapi-heimdal libssasl2-modules-ldap
  libssasl2-modules-otp libssasl2-modules-sql keychain libpam-ssh monkeysphere 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 git-man krb5-locales less libbrotli0 libbsd0 libcbor0.10
  libcurl3t64-gnutls libedit2 liberror-perl libexpat1 libfdio2-1 libgdbm-compat4t64 libgdbm6t64
  libgsapi-krb5-2 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2
  libnghttp2-14 libperl5.38t64 libpsl5t64 librmp1 libssasl2-2 libssasl2-modules libssasl2-modules-db
  libssh-4 libx11-6 libx11-data libxa6 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 [3110 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 kB]
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 [191 kB]
Get:7 http://archive.ubuntu.com/ubuntu/noble-updates/main amd64 openssl amd64 3.0.13-0ubuntu3.4 [1003 kB]
Get:8 http://archive.ubuntu.com/ubuntu/noble/main amd64 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 590-2ubuntu2.1 [142 kB]
```

Git installation.

7. Checking the git version using git –version.

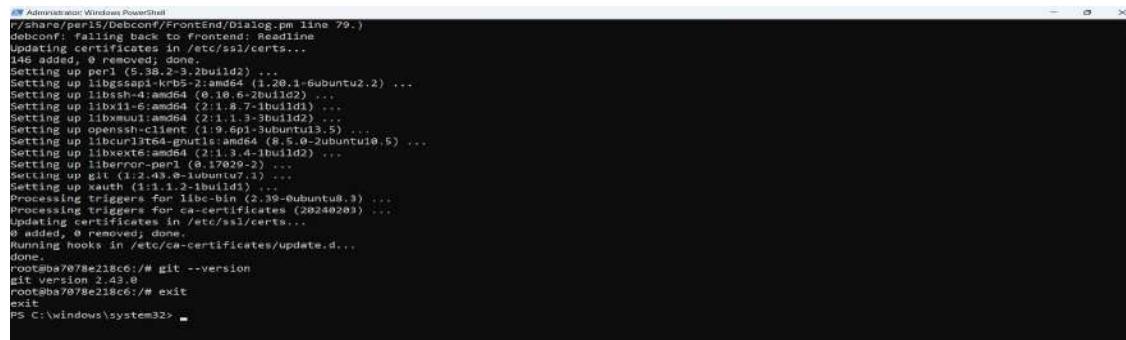


```

root@ba7078e218c6:/# debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 79.)
debconf: falling back to frontend: Readline
Updating certificates in /etc/ssl/certs...
146 added, 0 removed; done.
Setting up perl (5.38.2-3.2build2) ...
Setting up libgssapi-krb5-2:amd64 (1.20.1-6ubuntu2.2) ...
Setting up libssh-4:amd64 (0.10.6-2build2) ...
Setting up libx11-6:amd64 (2:1.8.7-1build1) ...
Setting up libmuu1:amd64 (2:1.1.3-3build2) ...
Setting up openssh-client (1:9.6p1-3ubuntu13.5) ...
Setting up libcurl3:amd64 (2:1.3.4-1build2) ...
Setting up libext2:amd64 (0.17029-2) ...
Setting up git (1:2.43.0-1ubuntu7.1) ...
Setting up xauth (1:1.1.2-1build1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@ba7078e218c6:/# git --version
git version 2.43.0
root@ba7078e218c6:/#

```

8. Exiting the git using exit command.

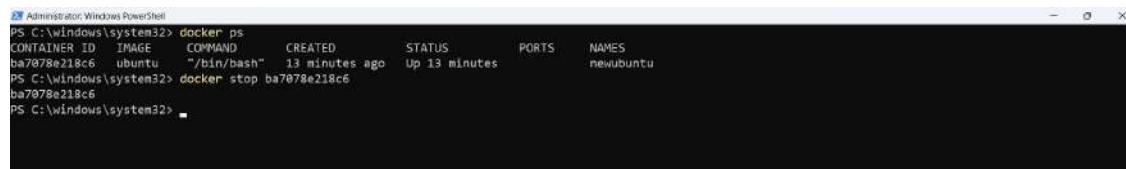


```

Administrator: Windows PowerShell
r\share\perl5\Debconf\FrontEnd\Dialog.pm line 79.)
debconf: falling back to frontend: Readline
Updating certificates in /etc/ssl/certs...
146 added, 0 removed; done.
Setting up perl (5.38.2-3.2build2) ...
Setting up libgssapi-krb5-2:amd64 (1.20.1-6ubuntu2.2) ...
Setting up libssh-4:amd64 (0.10.6-2build2) ...
Setting up libx11-6:amd64 (2:1.8.7-1build1) ...
Setting up libmuu1:amd64 (2:1.1.3-3build2) ...
Setting up openssh-client (1:9.6p1-3ubuntu13.5) ...
Setting up libcurl3:amd64 (2:1.3.4-1build2) ...
Setting up libext2:amd64 (2:1.3.4-1build2) ...
Setting up liberror-perl (0.17029-2) ...
Setting up git (1:2.43.0-1ubuntu7.1) ...
Setting up xauth (1:1.1.2-1build1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@ba7078e218c6:/# git --version
git version 2.43.0
root@ba7078e218c6:/# exit
exit
PS C:\Windows\system32>

```

9. Stopping the git container that we just started

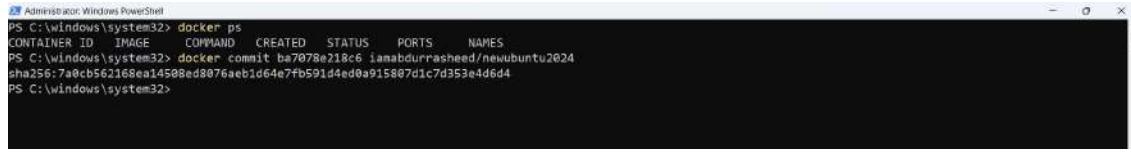


```

Administrator: Windows PowerShell
PS C:\Windows\system32> docker ps
CONTAINER ID        IMAGE             COMMAND            CREATED           STATUS            PORTS      NAMES
ba7078e218c6        ubuntu            "/bin/bash"        13 minutes ago   Up 13 minutes          newubuntu
PS C:\Windows\system32> docker stop ba7078e218c6
ba7078e218c6
PS C:\Windows\system32>

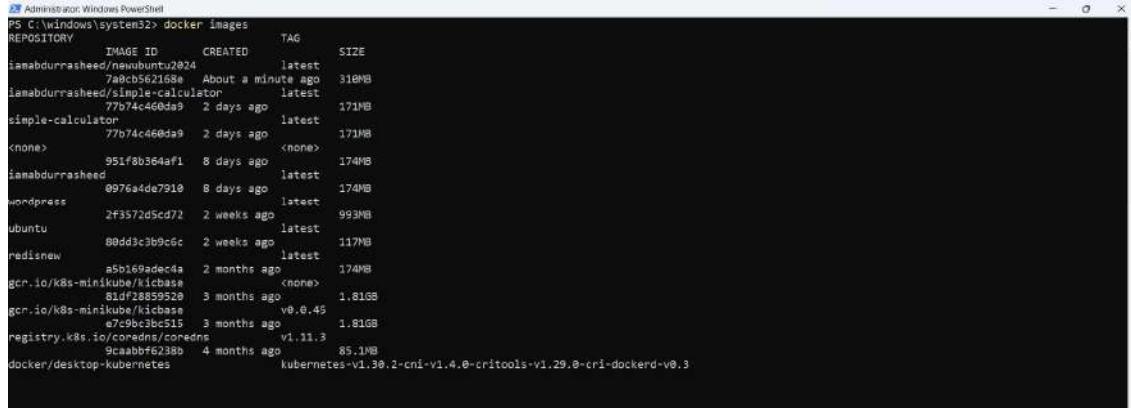
```

10. Commit the changes and saving the image.



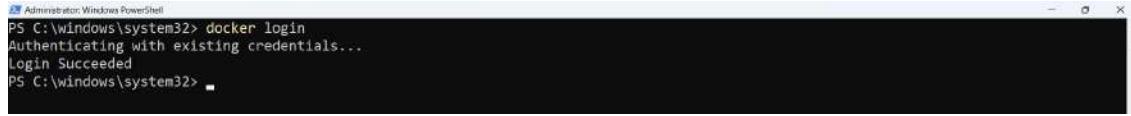
```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
PS C:\Windows\system32> docker commit ba7078e218c6 iamabdurrasheed/newubuntu:2024
sha256:7a0cb562168ea14508ed8076aeb1d64e7fb591d4ed0a915887d1c7d353e4d6d4
PS C:\Windows\system32>
```

11. Checking the docker images.



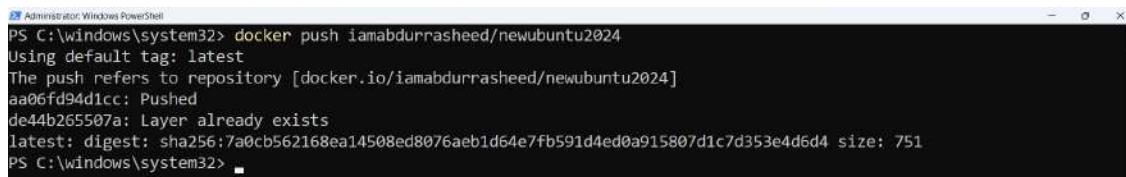
```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker images
REPOSITORY          IMAGE ID        CREATED         TAG           SIZE
iamabdurrasheed/newubuntu:2024   7a0cb562168e   About a minute ago   latest      310MB
iamabdurrasheed/simple-calculator 77974c460da9   2 days ago       latest      171MB
simple-calculator          77974c460da9   2 days ago       latest      171MB
<none>                951f80564af1   8 days ago       <none>     174MB
iamabdurrasheed          0976a4de7910   8 days ago       latest      174MB
wordpress              2f3572d5cd72   2 weeks ago      latest      993MB
ubuntu                 80dd3c3b9c6c   2 weeks ago      latest      117MB
redisnew               a5b169adec4a   2 months ago     latest      174MB
gcr.io/k8s-minikube/kibbase    81df28859520   3 months ago     <none>     1.81GB
gcr.io/k8s-minikube/kibbase    a79cfc1bc515   3 months ago     v0.0.45    1.81GB
registry.k8s.io/coredns      9caabfb6238b   4 months ago     v1.11.3    85.1MB
docker/desktop-kubernetes   kubernetes-v1.39.2-cni-v1.4.0-cri-tools-v1.29.0-cri-dockerd-v0.3
```

12. Login to the docker hub using docker login.



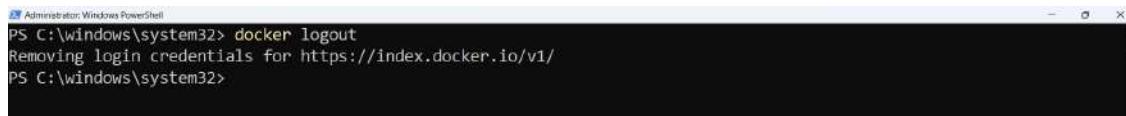
```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Windows\system32> .
```

13. Push the latest updated image into the docker hub.



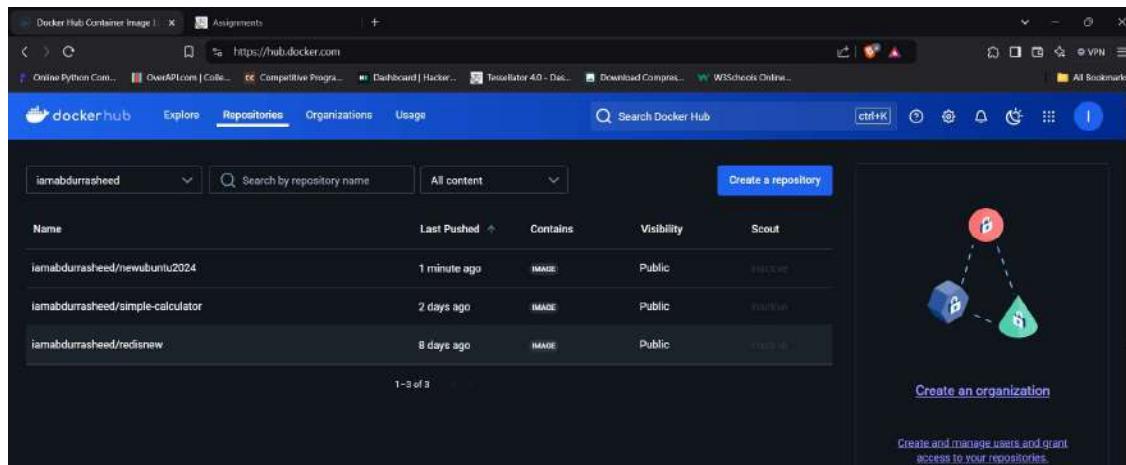
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker push iamabdurrasheed/newubuntu2024
Using default tag: latest
The push refers to repository [docker.io/iamabdurrasheed/newubuntu2024]
aa06fd94d1cc: Pushed
de44b265507a: Layer already exists
latest: digest: sha256:7a0cb562168ea14508ed8076aeb1d64e7fb591d4ed0a915807d1c7d353e4d6d4 size: 751
PS C:\windows\system32>
```

14. Logout from the docker hub.



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker logout
Removing login credentials for https://index.docker.io/v1/
PS C:\windows\system32>
```

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



16. Deleting the container locally .

```
PS C:\windows\system32> docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS
ba7078e218c6   ubuntu        "/bin/bash"   36 minutes ago   Exited (13)
7) 22 minutes ago   newubuntu
032552128e69   iamabdurrasheed/simple-calculator "docker-entrypoint.s..." 2 days ago   Exited (0)
2 days ago   fervent_clarke
a63679a9cce8   gcr.io/k8s-minikube/kicbase:v0.0.45  "/usr/local/bin/entr..."  12 days ago   Exited (13)
0) 2 hours ago   minikube
PS C:\windows\system32> docker rm ba7078e218c6
ba7078e218c6
PS C:\windows\system32>
```

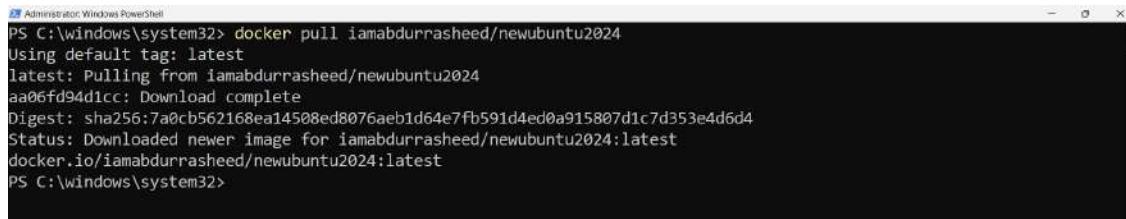
17. Deleting the image locally.

```
PS C:\windows\system32> docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS
ba7078e218c6   ubuntu        "/bin/bash"   36 minutes ago   Exited (13)
7) 22 minutes ago   newubuntu
032552128e69   iamabdurrasheed/simple-calculator "docker-entrypoint.s..." 2 days ago   Exited (0)
2 days ago   fervent_clarke
a63679a9cce8   gcr.io/k8s-minikube/kicbase:v0.0.45  "/usr/local/bin/entr..."  12 days ago   Exited (13)
0) 2 hours ago   minikube
PS C:\windows\system32> docker rm ba7078e218c6
ba7078e218c6
PS C:\windows\system32> docker rmi iamabdurrasheed/newubuntu2024
Untagged: iamabdurrasheed/newubuntu2024:latest
Deleted: sha256:7a0cb562168ea14508ed8076aebe1d64e7fb591d4ed0a915807d1c7d353e4d6d4
PS C:\windows\system32>
```

18. We can see the container is deleting.

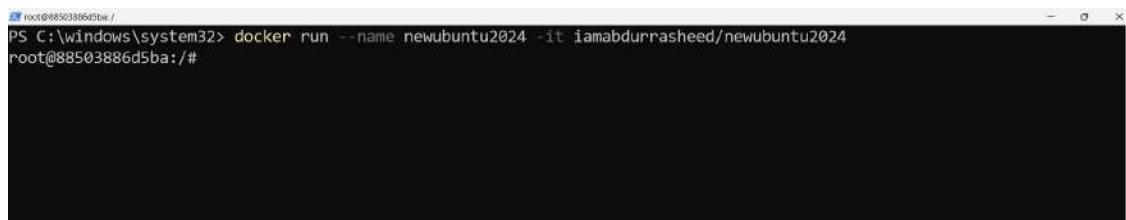
```
PS C:\windows\system32> docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS
032552128e69   iamabdurrasheed/simple-calculator "docker-entrypoint.s..."  2 days ago   Exited (0) 2
days ago   fervent_clarke
a63679a9cce8   gcr.io/k8s-minikube/kicbase:v0.0.45  "/usr/local/bin/entr..."  12 days ago   Exited (130)
2 hours ago   minikube
PS C:\windows\system32>
```

19. Pulling the image that we just pushed.



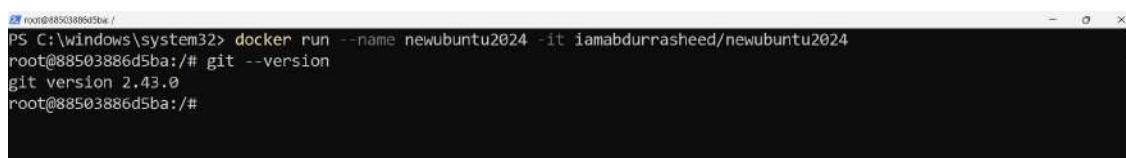
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker pull iamabdurrasheed/newubuntu2024
Using default tag: latest
latest: Pulling from iamabdurrasheed/newubuntu2024
aa06fd94diccc: Download complete
Digest: sha256:7a0cb562168ea14508ed8076aeb1d64e7fb591d4ed0a915807d1c7d353e4d6d4
Status: Downloaded newer image for iamabdurrasheed/newubuntu2024:latest
docker.io/iamabdurrasheed/newubuntu2024:latest
PS C:\windows\system32>
```

20. Running the image into a container.



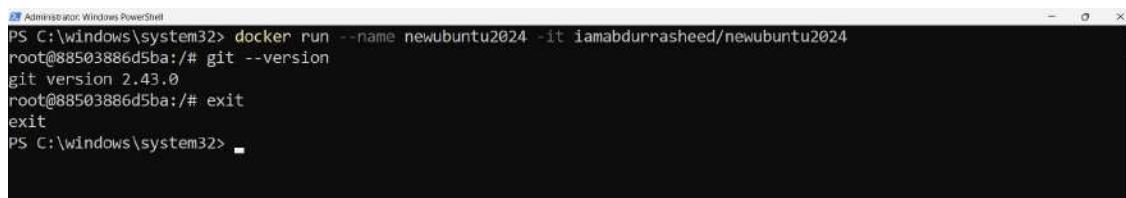
```
root@88503886d5ba:/#
PS C:\windows\system32> docker run --name newubuntu2024 -it iamabdurrasheed/newubuntu2024
root@88503886d5ba:/#
```

21. Checking the git version and making sure that the same git version is seen.



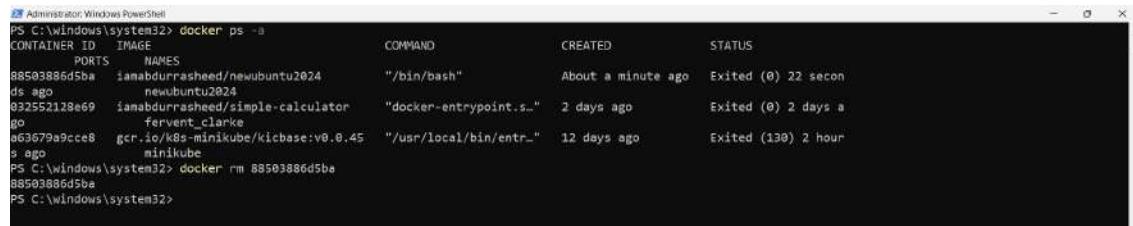
```
root@88503886d5ba:/#
PS C:\windows\system32> docker run --name newubuntu2024 -it iamabdurrasheed/newubuntu2024
root@88503886d5ba:/# git --version
git version 2.43.0
root@88503886d5ba:/#
```

22. Exiting the Git.



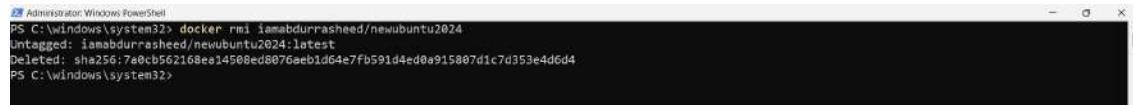
```
root@88503886d5ba:/#
PS C:\windows\system32> docker run --name newubuntu2024 -it iamabdurrasheed/newubuntu2024
root@88503886d5ba:/# git --version
git version 2.43.0
root@88503886d5ba:/# exit
exit
PS C:\windows\system32>
```

23. Deleting the container that we pulled from docker hub locally.



```
Administrator:Windows PowerShell
PS C:\Windows\system32> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
88503886d5ba      iamabdurrasheed/newubuntu2024   "/bin/bash"        About a minute ago   Exited (0) 22 secon
ds ago             newubuntu2024
032552128e69      iamabdurrasheed/simple-calculator  "docker-entrypoint.s..."  2 days ago       Exited (0) 2 days a
go                fervent_clarke
063679a9cce8      gcr.io/k8s-minikube/kicbase:v0.0.45    "/usr/local/bin/entr..."  12 days ago      Exited (130) 2 hour
s ago              minikube
PS C:\Windows\system32> docker rm 88503886d5ba
88503886d5ba
PS C:\Windows\system32>
```

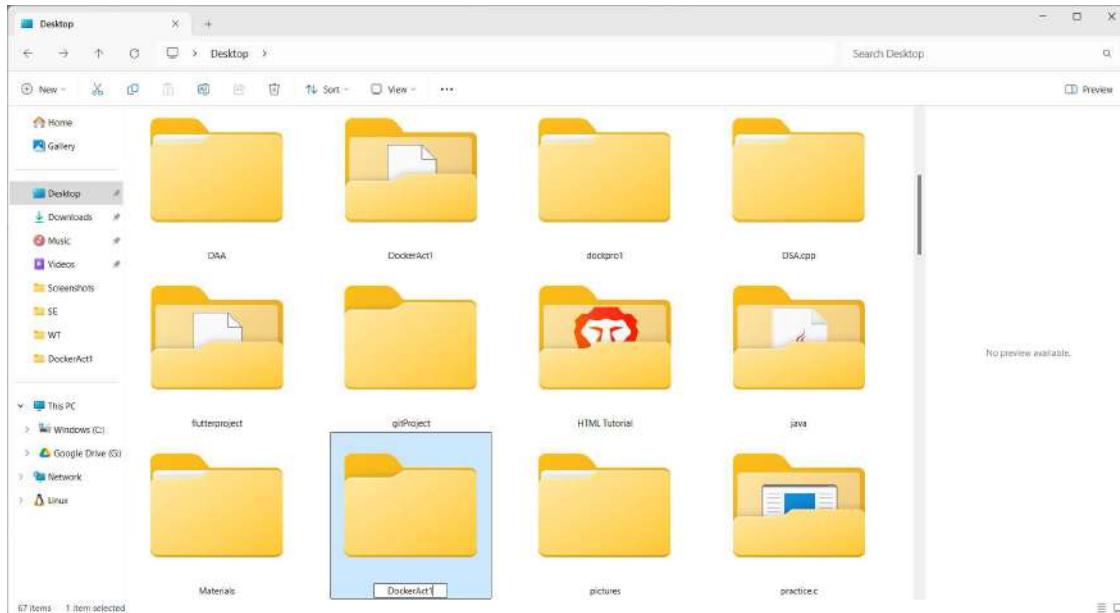
24. Deleting the image that we pulled from docker hub locally



```
Administrator:Windows PowerShell
PS C:\Windows\system32> docker rmi iamabdurrasheed/newubuntu2024
Untagged: iamabdurrasheed/newubuntu2024:latest
Deleted: sha256:7a0cb562168ea14568ed8076aeb1d64e7fb591d4ed0a915807d1c7d353e4d6d4
PS C:\Windows\system32>
```

7 D. CREATE AND PUSH DOCKER FILE IMAGE

Create a Folder named dockerCalculator.



Using command prompt create a file with no extension

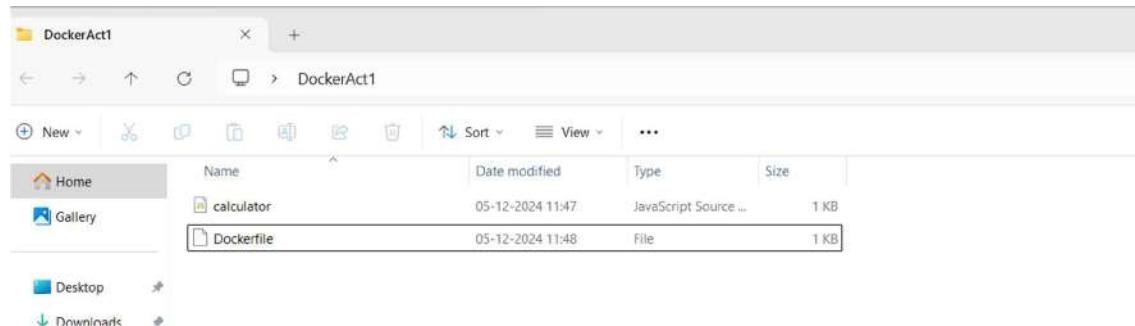
```
C:\windows\system32\cmd.exe > + <
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abdur>cd C:\Users\abdur\OneDrive\Desktop\DockerAct1

C:\Users\abdur\OneDrive\Desktop\DockerAct1>echo. > Dockerfile

C:\Users\abdur\OneDrive\Desktop\DockerAct1>
```

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



DockerFile and Calulator.js file...

```
calculator.js > ...
JS calculator.js > ...
1 // calculator.js
2 // Function to add two numbers
  CodeLens Refactor | Explain | X
3 function add(a, b) {
4   return a + b;
5 }
6 // Function to subtract two numbers
  CodeLens Refactor | Explain | X
7 function subtract(a, b) {
8   return a - b;
9 }
10 // Function to multiply two numbers
  CodeLens Refactor | Explain | X
11 function multiply(a, b) {
12   return a * b;
13 }
14 // Function to divide two numbers
  CodeLens Refactor | Explain | X
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));
```



A screenshot of a Dockerfile editor interface. The title bar says "Dockerfile". The menu bar includes "File", "Edit", and "View". The main area contains the following Dockerfile code:

```
FROM node:16-alpine
WORKDIR /app
COPY calculator.js /app
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.

Running the Calculator image.

```
C:\Windows\system32\cmd.exe > C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker run simple-calculator
Addition (2 + 3): 5
Subtraction (5 - 2): 3
Multiplication (4 * 3): 12
Division (10 / 2): 5

C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS          PORTS     NAMES
C:\Users\abdur\OneDrive\Desktop\DockerAct1>
```

Login to docker hub to push the image.

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker login
Authenticating with existing credentials...
Login Succeeded
```

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>
```

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

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker tag simple-calculator iamabdurrasheed/simple-calculator
```

Pushing the image into my docker hub .

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker push iamabdurrasheed/simple-calculator
Using default tag: latest
The push refers to repository [docker.io/iamabdurrasheed/simple-calculator]
7264a8db6415: Pushed
5f4e04e05d4b: Pushed
fcd28dec93bd: Pushed
eee371b9ce3f: Pushed
d9059661ce70: Pushed
93b3025fe103: Pushed
cccd67ccaaad3: Pushed
latest: digest: sha256:77b74c460da99a547775e47351d15dcd2fe979c040cb28df9ee90c408c65402c size: 856
C:\Users\abdur\OneDrive\Desktop\DockerAct1>
```

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

Name	Last Pushed	Contains	Visibility	Scout
iamabdurrasheed/simple-calculator	2 days ago	IMAGE	Public	Inactive
iamabdurrasheed/redisnew	8 days ago	IMAGE	Public	Inactive

Removing the container locally: The image shows the docker ps -a command listing all Docker containers, followed by the command docker rm used to remove a specific container by its ID (3107aef5231b).

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
3107aef5231b        simple-calculator   "docker-entrypoint.s..."   6 minutes ago    Exited (0) 6 minutes ago   elated_pascal
a63079a9cc8          gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr..."   10 days ago      Exited (255) 7 days 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
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker rm 3107aef5231b
3107aef5231b
```

Removing the image locally: The command docker rmi is used to delete the Docker image iamaburrasheed/simple-calculator:latest, with a confirmation of the image being untagged and removed.

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker rmi iamaburrasheed/simple-calculator
Untagged: iamaburrasheed/simple-calculator:latest
```

Pulling the latest image from Docker Hub: The docker pull command retrieves the iamaburrasheed/simple-calculator image from Docker Hub, showing details such as the digest and status of the download.

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker pull iamaburrasheed/simple-calculator
Using default tag: latest
latest: Pulling from iamaburrasheed/simple-calculator
Digest: sha256:77b74c460da99a547775e47351d15dc2fe979c040cb28df9ee90c408c65402c
Status: Downloaded newer image for iamaburrasheed/simple-calculator:latest
docker.io/iamaburrasheed/simple-calculator:latest
```

Running the pulled image: The docker run command executes the iamaburrasheed/simple-calculator image, performing various calculations (addition, subtraction, multiplication, and division) with the output displayed in the terminal.

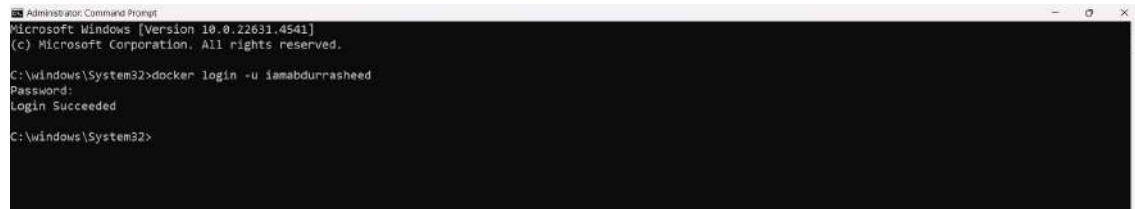
```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker run iamaburrasheed/simple-calculator
Addition (2 + 3): 5
Subtraction (5 - 2): 3
Multiplication (4 * 3): 12
Division (10 / 2): 5
```

Deleting the container: The docker logout command logs the user out of Docker Hub, removing stored login credentials for the repository URL.

```
C:\Users\abdur\OneDrive\Desktop\DockerAct1>docker logout
Removing login credentials for https://index.docker.io/v1/
```

7 E. RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE

Login to docker using command prompt.

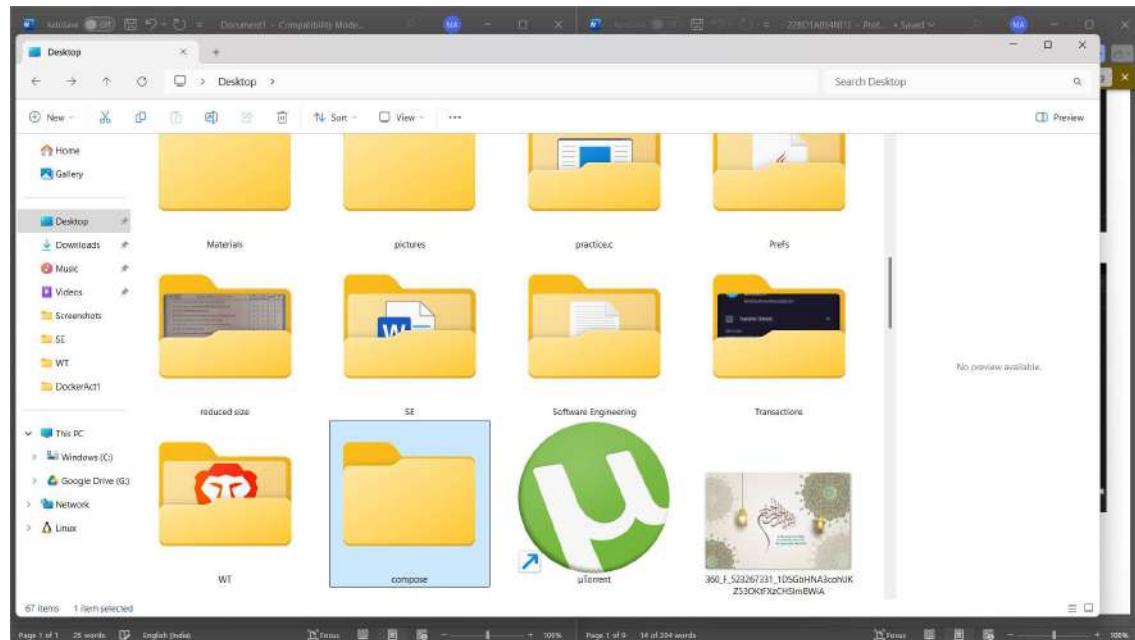


```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

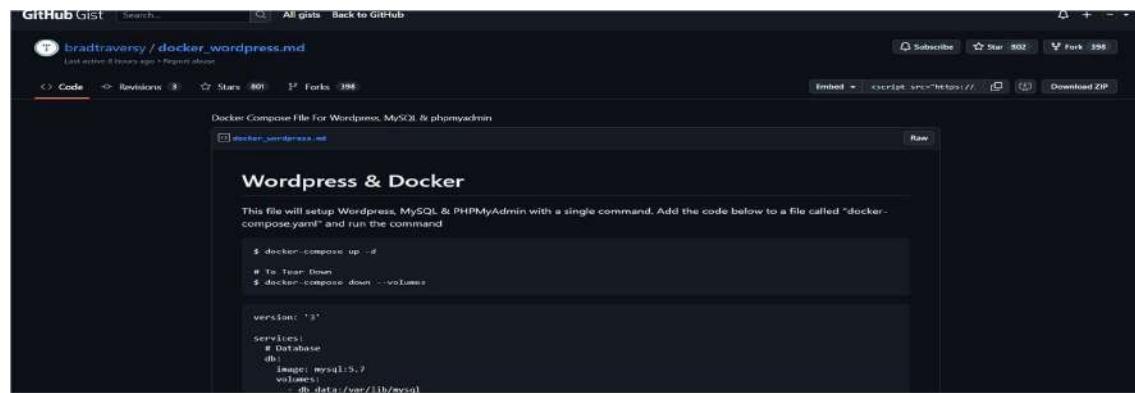
C:\Windows\System32>docker login -u iamabdurrasheed
Password:
Login Succeeded

C:\Windows\System32>
```

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



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



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

```

1  services:
2    # Database
3    db:
4      image: mysql:5.7
5      volumes:
6        - db_data:/var/lib/mysql
7      restart: always
8      environment:
9        MYSQL_ROOT_PASSWORD: password
10       MYSQL_DATABASE: wordpress
11       MYSQL_USER: wordpress
12       MYSQL_PASSWORD: wordpress
13      networks:
14        - wpsite
15    # phpmyadmin
16    phpmyadmin:
17      depends_on:
18        - db
19      image: phpmysqladmin/phpmyadmin
20      restart: always
21      ports:
22        - '8080:80'
23      environment:
24        PMA_HOST: db
25        MYSQL_ROOT_PASSWORD: password
26      networks:
27        - wpsite
28    # Wordpress
29    wordpress:
30      depends_on:
31        - db
32      image: wordpress:latest
33      ports:
34        - '8000:80'
35      restart: always
36      volumes: ['./var/www/html']
37      environment:
38        WORDPRESS_DB_HOST: db:3306
39        WORDPRESS_DB_USER: wordpress
40        WORDPRESS_DB_PASSWORD: wordpress
41      networks:
42        - wpsite
43      networks:
44        - wpsite
45      volumes:
46        - db_data:

```

Navigate to the folder that we just created.

```

Administrator: Command Prompt
C:\windows\System32>cd C:\Users\abdur\OneDrive\Desktop\compose
C:\Users\abdur\OneDrive\Desktop\compose>

```

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

```

Administrator: Command Prompt
C:\windows\System32>cd C:\Users\abdur\OneDrive\Desktop\compose
C:\Users\abdur\OneDrive\Desktop\compose>docker-compose --version
Docker Compose version v2.29.7-desktop.1
C:\Users\abdur\OneDrive\Desktop\compose>

```

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

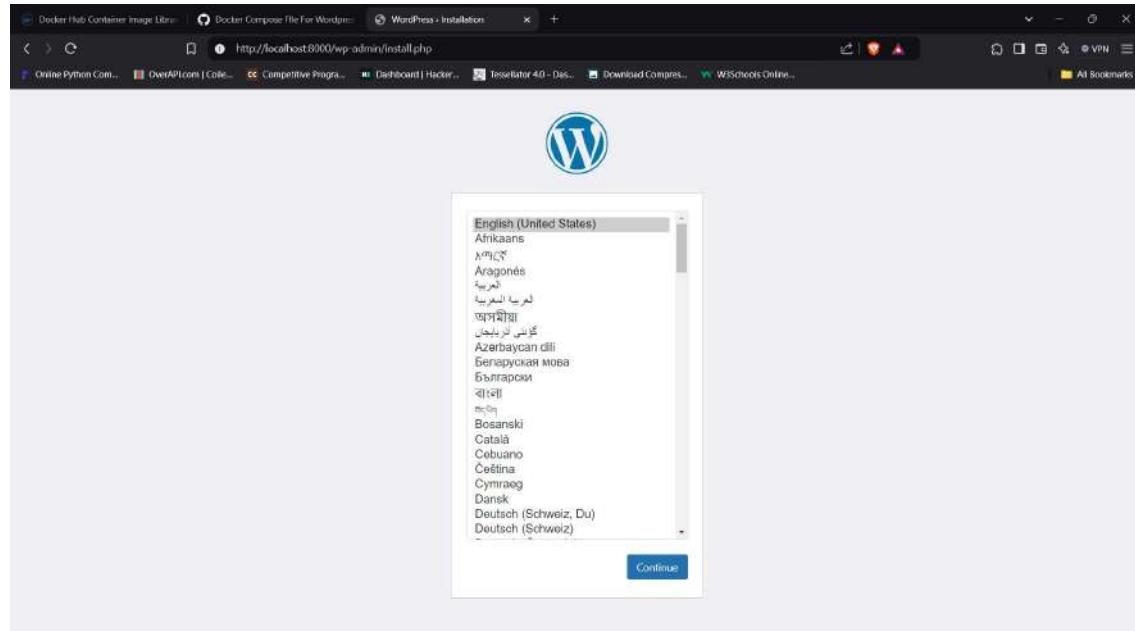
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\compose> docker-compose up -d
[+] Running 4/2
  ✓ wordpress Pulled
  ✓ phpmyadmin Pulled
  ✓ db Pulled

[+] Running 4/5
  ✓ Network compose_website      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 0.0.0.0:8000 -> 0.0.0.0:0: listen tcp4 0.0.0.0:8000: bind: An attempt was made to access a socket in a way forbidden by its access permissions.
PS C:\Users\abdur\OneDrive\Desktop\compose>
```

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

```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\compose> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS          NAMES
fa0572290955   wordpress:latest   "docker-entrypoint.s..."   About a minute ago   Up About a minute   0.0.0.0:8000->80/tcp   compose_wordpress-1
f069234307e2   phpmyadmin/phpmyadmin   "/docker-entrypoint.s..."   About a minute ago   Up About a minute   3306/tcp, 33060/tcp   compose_phpmyadmin-1
87d52ee02498   mysql:5.7           "docker-entrypoint.s..."   About a minute ago   Up About a minute   3306/tcp, 33060/tcp   compose_db-1
032552128e69   iamabdurusshed/simple-calculator   "/docker-entrypoint.s..."   47 hours ago      Exited (0) 47 hours ago   3306/tcp, 33060/tcp   fervent_minikube
a6367989cc8    gcr.io/k8s-minikube/kicbase:v0.45   "/usr/local/bin/entr..."   12 days ago      Exited (130) 24 minutes ago   minikube
PS C:\Users\abdur\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.

The screenshot shows a web browser window titled "Docker Hub Container Image Library" with the URL "http://localhost:8000/wp-admin/install.php?step=1". The page is titled "Welcome" and contains the following form fields:

Site Title	iamabdurrasheed
Username	iamabdurrasheed
Password	***** Medium
Your Email	abdurrasheed350@gmail.com
Search engine visibility	<input type="checkbox"/> Discourage search engines from indexing this site It is up to search engines to honor this request.

At the bottom of the form is a blue "Install WordPress" button.

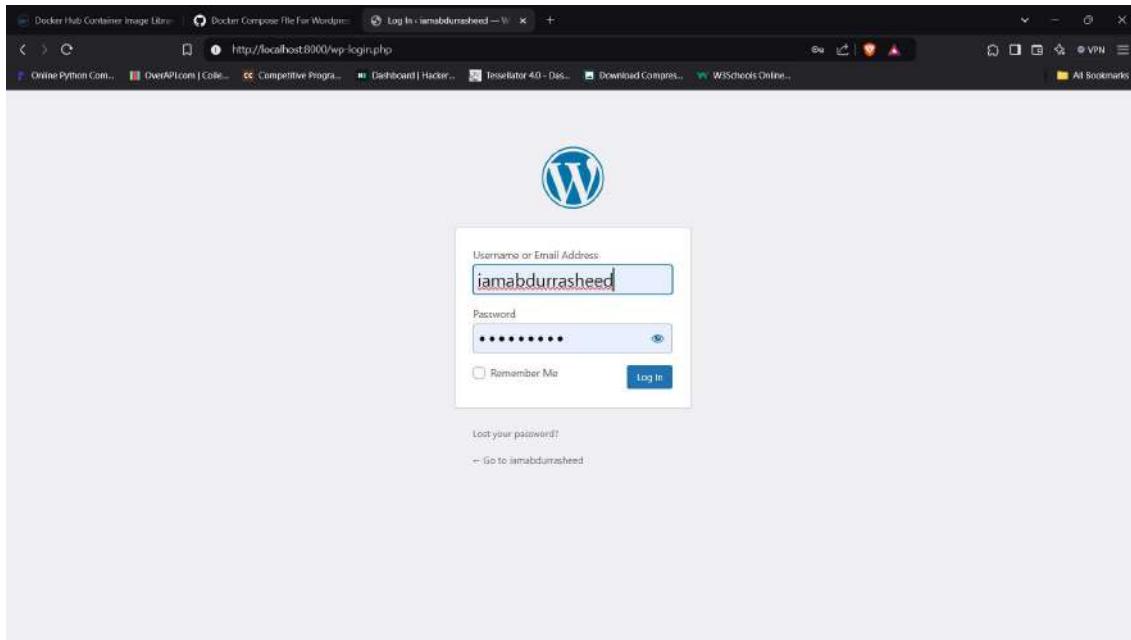
Continue to login.

The screenshot shows a web browser window titled "Docker Hub Container Image Library" with the URL "http://localhost:8000/wp-admin/install.php?step=2". The page features a large blue "W" logo at the top. Below it, the word "Success!" is displayed in a box. The message "WordPress has been installed. Thank you, and enjoy!" is shown. The user's credentials are listed:

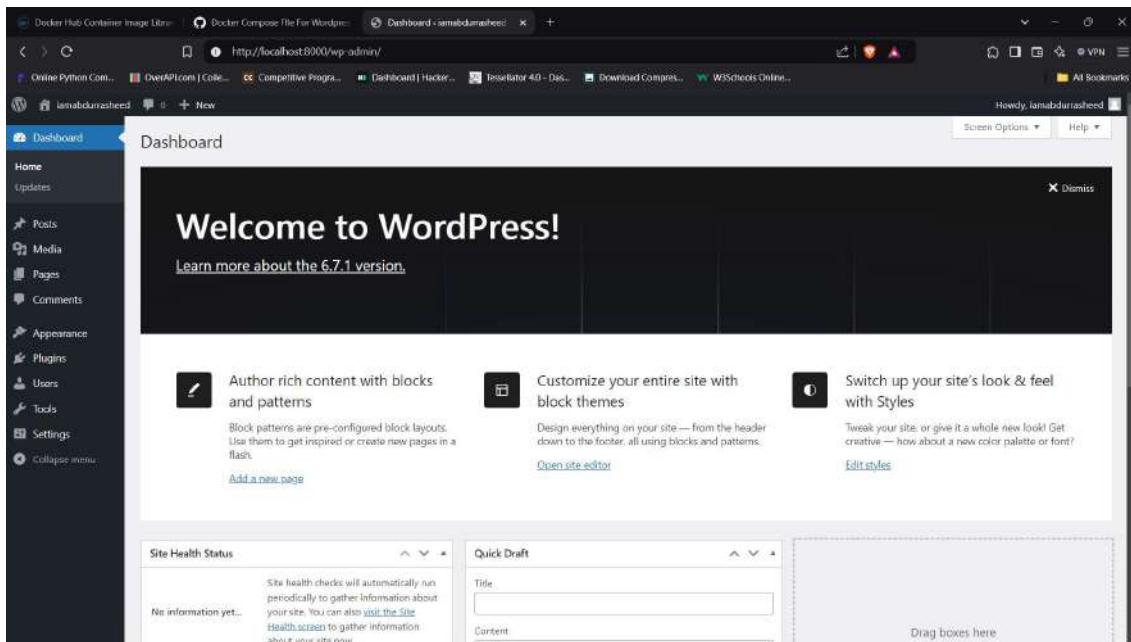
Username	iamabdurrasheed
Password	Your chosen password.

A green "Log In" button is located at the bottom of the success message box.

Login with the credentials that you just provided.



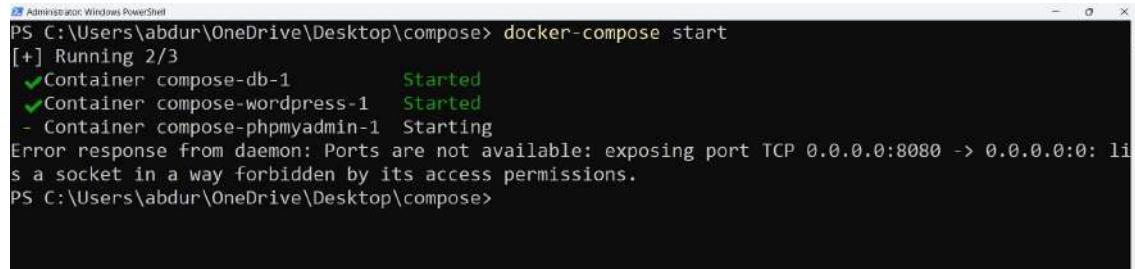
We have successfully created the account in wordpress inside the docker.



Stop the compose using docker-compose stop.

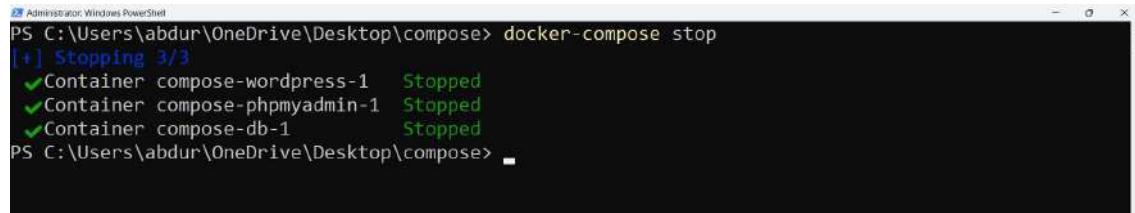
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\compose> docker-compose stop
[+] Stopping 3/3
✓ Container compose-phpmyadmin-1 Stopped
✓ Container compose-wordpress-1 Stopped
✓ Container compose-db-1 Stopped
PS C:\Users\abdur\OneDrive\Desktop\compose>
```

Starting the compose using docker-compose start.



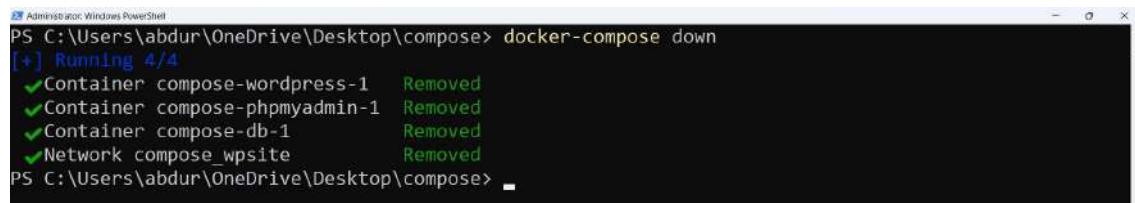
```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\compose> docker-compose start
[+] Running 2/3
  ✓ Container compose-db-1      Started
  ✓ Container compose-wordpress-1 Started
  - Container compose-phpmyadmin-1 Starting
Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:8080 -> 0.0.0.0:0: listening on a socket in a way forbidden by its access permissions.
PS C:\Users\abdur\OneDrive\Desktop\compose>
```

Stop the compose using docker-compose stop.



```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\compose> docker-compose stop
[+] Stopping 3/3
  ✓ Container compose-wordpress-1 Stopped
  ✓ Container compose-phpmyadmin-1 Stopped
  ✓ Container compose-db-1 Stopped
PS C:\Users\abdur\OneDrive\Desktop\compose>
```

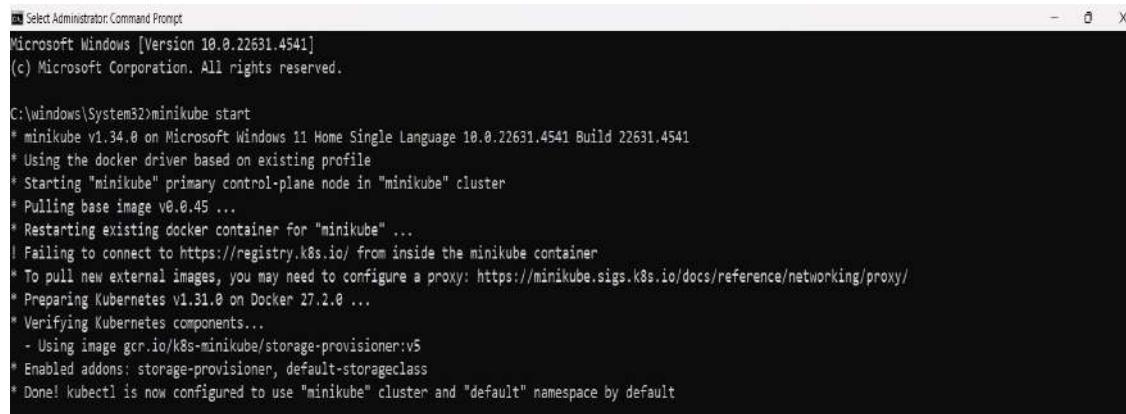
Deleting/Removing the compose using docker-compose down.



```
Administrator: Windows PowerShell
PS C:\Users\abdur\OneDrive\Desktop\compose> docker-compose down
[+] Running 4/4
  ✓ Container compose-wordpress-1 Removed
  ✓ Container compose-phpmyadmin-1 Removed
  ✓ Container compose-db-1 Removed
  ✓ Network compose_wpsite Removed
PS C:\Users\abdur\OneDrive\Desktop\compose>
```

7 F. DEPLOYING AND SCALING APPLICATIONS USING MINIKUBE

Start Minikube: This image shows the process of starting Minikube, a local Kubernetes cluster, using the minikube start command. Details of the cluster configuration, such as Kubernetes version and namespace setup, are displayed. Minikube provides a way to test Kubernetes deployments locally.



```
Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\windows\System32>minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4541 Build 22631.4541
* 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 inside the minikube container
* 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: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

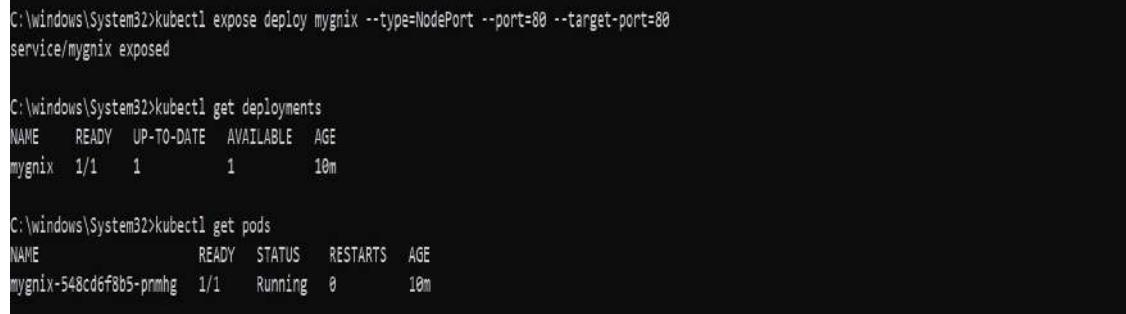
Create the Nginx Deployment: The kubectl create deployment mynginx --image=nginx command creates a Kubernetes deployment for the Nginx server. The deployment uses the Nginx container image, which is confirmed by the kubectl get deployments output listing the created deployment.



```
C:\windows\System32>kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created

C:\windows\System32>kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx  0/1     1           0           16s
```

Expose the Nginx Deployment to the Outside World: Using kubectl expose deploy mynginx --type=NodePort --port=80 --target-port=80, the Nginx deployment is exposed to external traffic. The kubectl get services command confirms the creation of the service with a NodePort for accessing the application.



```
C:\windows\System32>kubectl expose deploy mynginx --type=NodePort --port=80 --target-port=80
service/mynginx exposed

C:\windows\System32>kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx  1/1     1           1           10m

C:\windows\System32>kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
mynginx-548cd6f8b5-prmhg  1/1     Running   0          10m
```

Scale the Deployment: The deployment is scaled to multiple replicas using the kubectl scale deployment nginx --replicas=4 command. The output of kubectl get pods shows the creation of four pods running the Nginx server, ensuring high availability and load balancing.

```
C:\windows\System32>kubectl scale deployment mygnix --replicas=4
deployment.apps/mygnix scaled

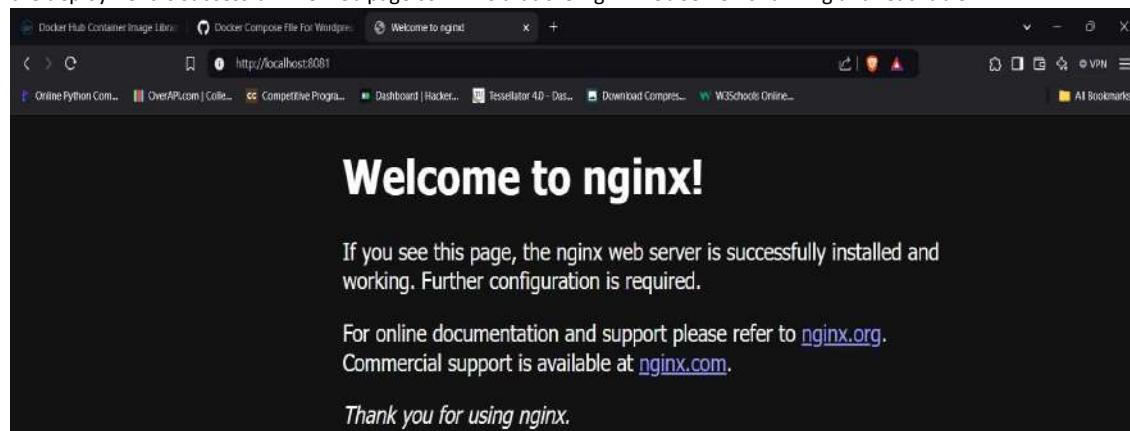
C:\windows\System32>kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
mygnix-548cd6f8b5-bvqrg  1/1     Running   0          76s
mygnix-548cd6f8b5-prnmg  1/1     Running   0          13m
mygnix-548cd6f8b5-rkhhb  1/1     Running   0          76s
mygnix-548cd6f8b5-wv2h  1/1     Running   0          76s

C:\windows\System32>kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mygnix   4/4       4           4           17m
```

Accessing the Nginx App: The kubectl port-forward svc/nginx 8081:80 command forwards port 8081 on the local machine to the Nginx service port (80). This allows access to the Nginx app locally, confirming its functionality.

```
C:\windows\System32>kubectl port-forward svc/mygnix 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 go to <http://localhost:8081>: The browser displays the default Nginx welcome page, indicating that the deployment is successful. The web page confirms that the Nginx web server is running and reachable.



Stopping and Cleaning Up Everything: The kubectl delete service nginx and kubectl delete deployment nginx commands are used to delete the Nginx service and deployment. Additionally, the minikube stop command shuts down the Minikube cluster, cleaning up resources.

```
C:\windows\System32>kubectl delete service mygnix
service "mygnix" deleted

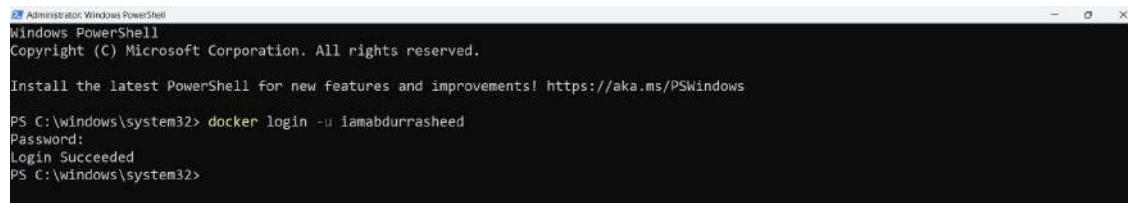
C:\windows\System32>kubectl delete deployment mygnix
deployment.apps "mygnix" deleted

C:\windows\System32>minikube stop
* Stopping node "minikube" ...
* Powering off "minikube" via SSH ...
* 1 node stopped.

C:\windows\System32>
```

7 G. DEPLOYING AND MANAGING MONITORING SYSTEMS USING NAGIOS IN DOCKER

Open power shell or cmd prompt in administrator mode and login to docker.

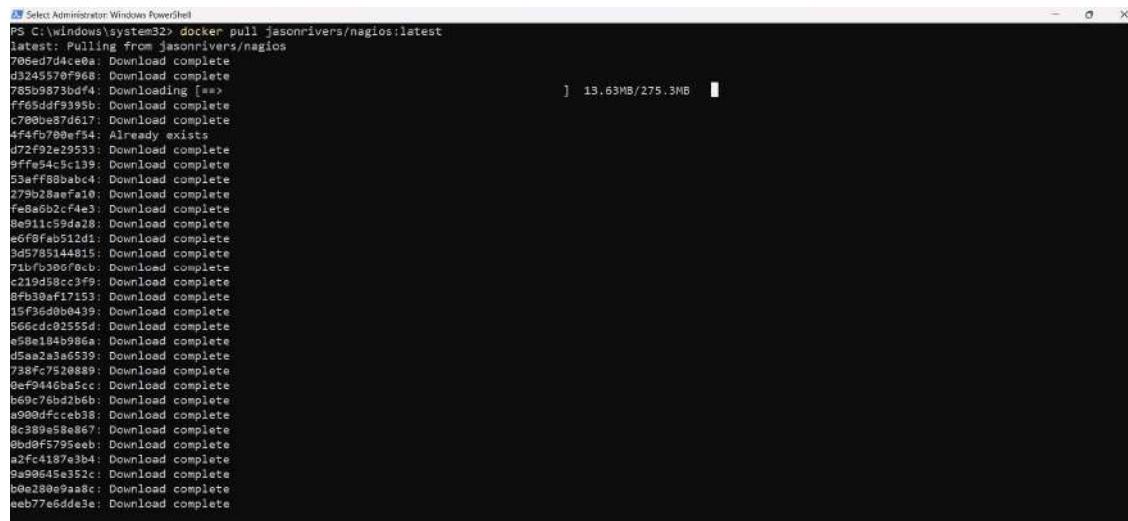


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

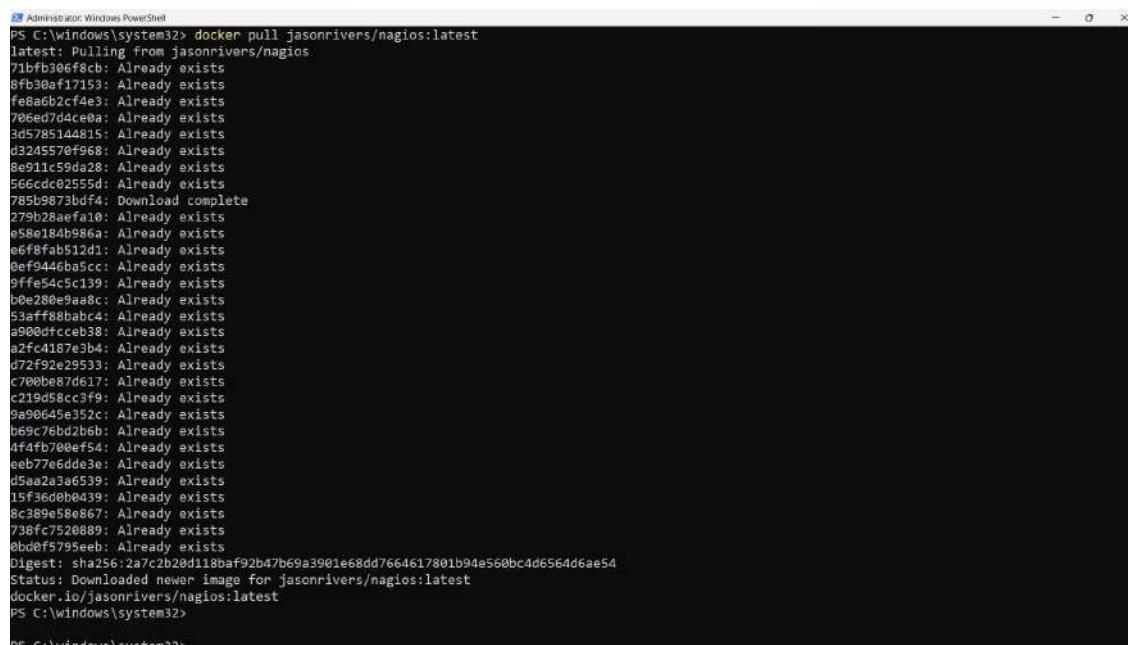
PS C:\windows\system32> docker login -u iamabdurrasheed
Password:
Login Succeeded
PS C:\windows\system32>
```

Pull the nagios latest image.



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
708edc7d4c:e0a: Download complete
d3245570f968: Download complete
785b0873bd74: Downloading [==>]
/ff65ddff9399b: Download complete
c700be87d617: Download complete
4f4fb780ef54: Already exists
d72f92e29533: Download complete
effe44c5c139: Download complete
53afff809abc4: Download complete
279b28aeefaf10: Download complete
fe8ab6b2c4e3: Download complete
8e911c59da28: Download complete
e6f8fbaf512d1: Download complete
3d5785144815: Download complete
71bf7b306f8cb: Download complete
c219d58cc3f9: Download complete
8fb30aef17153: Download complete
15f36d0b0439: Download complete
566cdcd02555d: Download complete
e58e184b586a: Download complete
d5aa2a3a6539: Download complete
738fc7520889: Download complete
0ef9446ba5cc: Download complete
b69c76bd2b6b: Download complete
a990dfccceb38: Download complete
8c389e58e867: Download complete
6008f5795eeb8: Download complete
a2fc4187e3b4: Download complete
9a99c45e52c: Download complete
b0e280e9aa8c: Download complete
eab77e6dde3e: Download complete
```

Wait for the image to be pulled.



```
Administrator: Windows PowerShell
PS C:\windows\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
71fb3aef8cb: Already exists
8fb30aef17153: Already exists
fe8ab6b2c4e3: Already exists
708edc7d4ce0a: Already exists
3d5785144815: Already exists
d3245570f968: Already exists
8e911c59da28: Already exists
566cdcd02555d: Already exists
785b9873bd74: Download complete
279b28aeefaf10: Already exists
e58e184b586a: Already exists
e6f8fab512d1: Already exists
0ef9446ba5cc: Already exists
9ffe54c5c139: Already exists
b0e280e9aa8c: Already exists
53afff809abc4: Already exists
a990dfccceb38: Already exists
a2fc4187e3b4: Already exists
d72f92e29533: Already exists
c700be87d617: Already exists
c219d58cc3f9: Already exists
9a99c45e52c: Already exists
b69c76bd2b6b: Already exists
4f4fb780ef54: Already exists
eab77e6dde3e: Already exists
d5aa2a3a6539: Already exists
15f36d0b0439: Already exists
8c389e58e867: Already exists
738fc7520889: Already exists
0bd0f5795eeb8: Already exists
Digest: sha256:2a7c2b20d118ba92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
Status: Downloaded newer image for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest
PS C:\windows\system32>
```

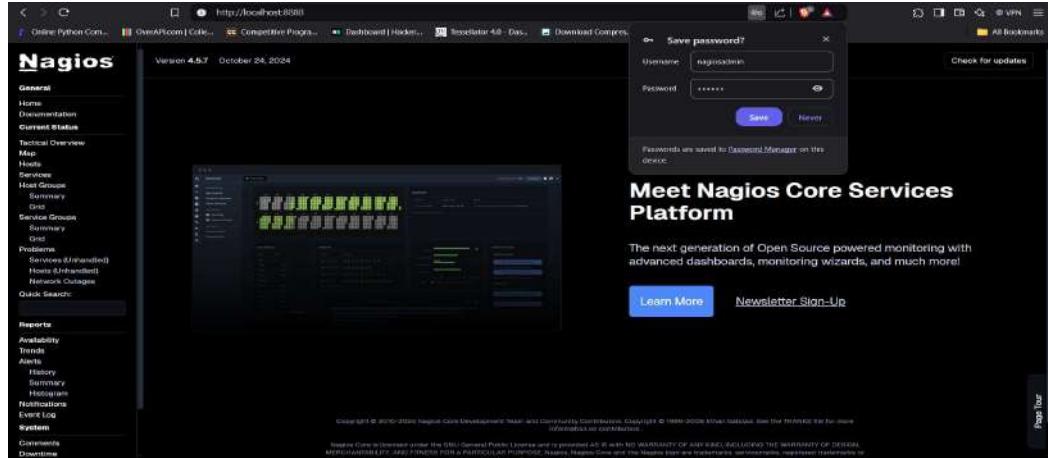
Now Run the docker image in a container

We can see the Nagios image is running in the localhost:8888 that we have provided above.

Go to google and open <http://localhost:8888> and Login to nagios admin using the username:nagiosadmin, password:nagios.



The Nagios Desktop



The Hosts Dashboard that Nagios is monitoring on our computer.

Current Network Status

Last Updated: Wed Dec 11 07:24:14 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.7 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

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

Service Status Totals

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

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
localhost	UP	12-11-2024 07:22:17	0d 0h 0m 3s	PING OK - Packet loss = 0%, RTA = 0.08 ms

Results 1 - 1 of 1 Matching Hosts

More Details about localhost.

Host Information

Last Updated: Wed Dec 11 07:24:35 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.7 - www.nagios.org
Logged in as nagiosadmin

Host
localhost
(localhost)

Member of
linux-servers

127.0.1

Host State Information

Host Information:	UP (for 0d 0h 0m 24s)
Performance Data:	rtt=0.046000ms,0.000000000.5000000.0000000.0000000 ph=0%,80,100,0
Current Attempt:	1/10 (HARD state)
Last Check Time:	12-11-2024 07:22:17
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4174 seconds
Next Scheduled Active Check:	12-11-2024 07:27:17
Last State Change:	12-11-2024 07:15:11
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	12-11-2024 07:24:29 (0d 0h 0m 6s ago)

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable event handler for this host
- Disable flap detection for this host
- Clear flapping state for this host

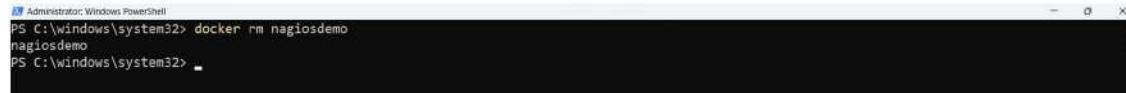
Close and reopen windows powershell as administrator to stop Nagios container

```
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

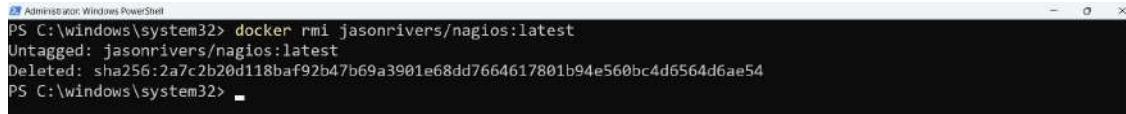
PS C:\windows\system32> docker stop nagiosdemo
nagiosdemo
PS C:\windows\system32>
```

Now remove the Nagios container and the Nagios image that we pulled.



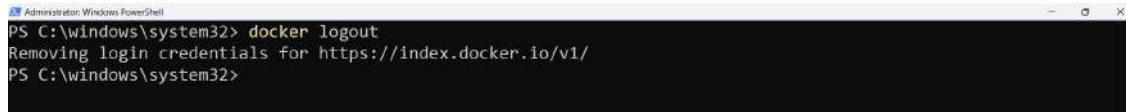
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker rm nagiosdemo
nagiosdemo
PS C:\windows\system32>
```

Now remove the Nagios image that we pulled.



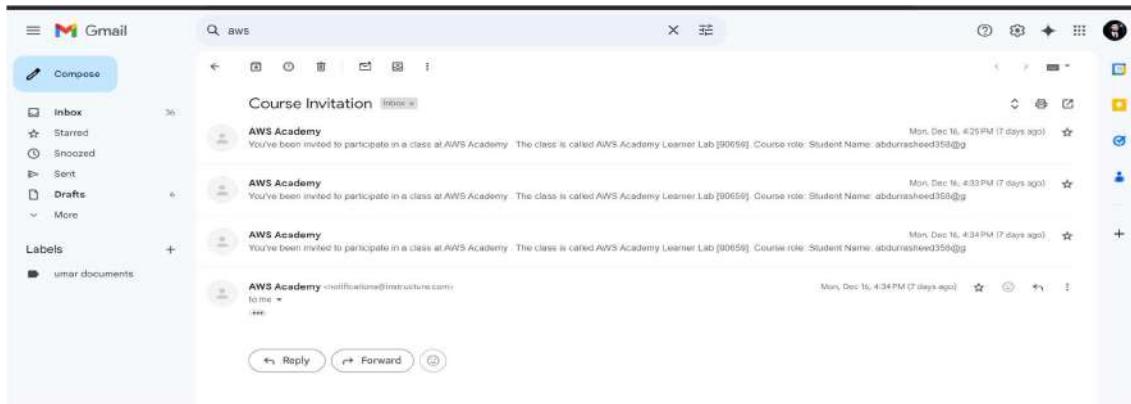
```
Administrator: Windows PowerShell
PS C:\windows\system32> docker rmi jasonrivers/nagios:latest
Untagged: jasonrivers/nagios:latest
Deleted: sha256:2a7c2b20d118baf92b47b69a3901e68dd7664617801b94e560bc4d6564d6ae54
PS C:\windows\system32>
```

Logging out from the Docker.

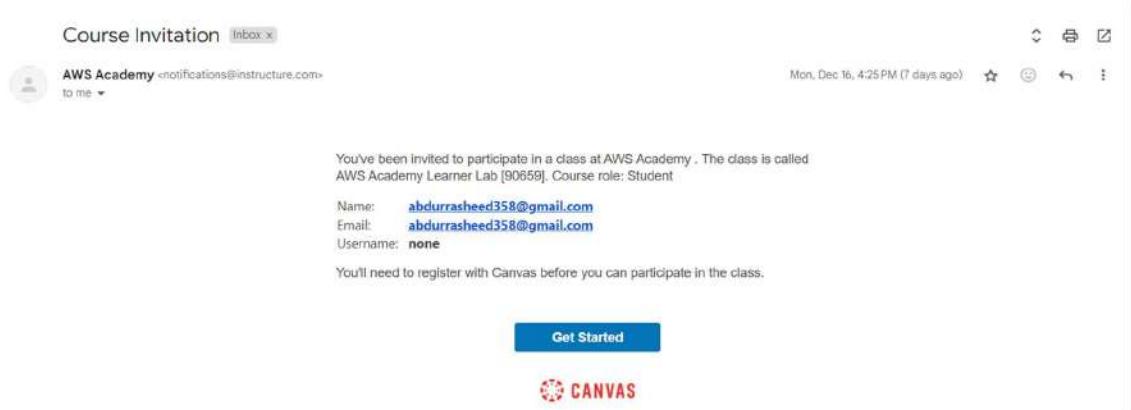


```
Administrator: Windows PowerShell
PS C:\windows\system32> docker logout
Removing login credentials for https://index.docker.io/v1/
PS C:\windows\system32>
```

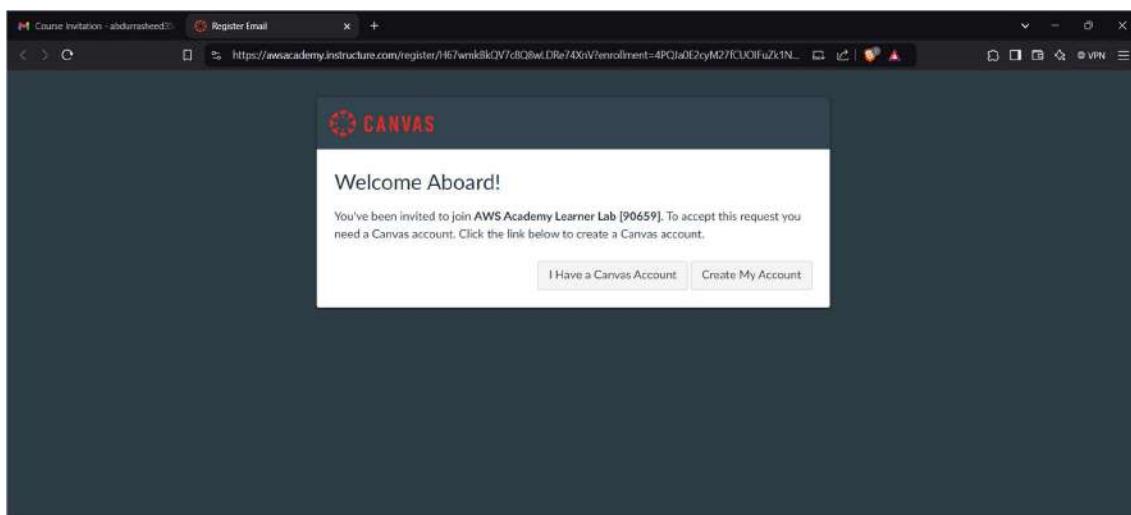
8 A. AWS ACADEMY LEARNING ACCOUNT CREATION



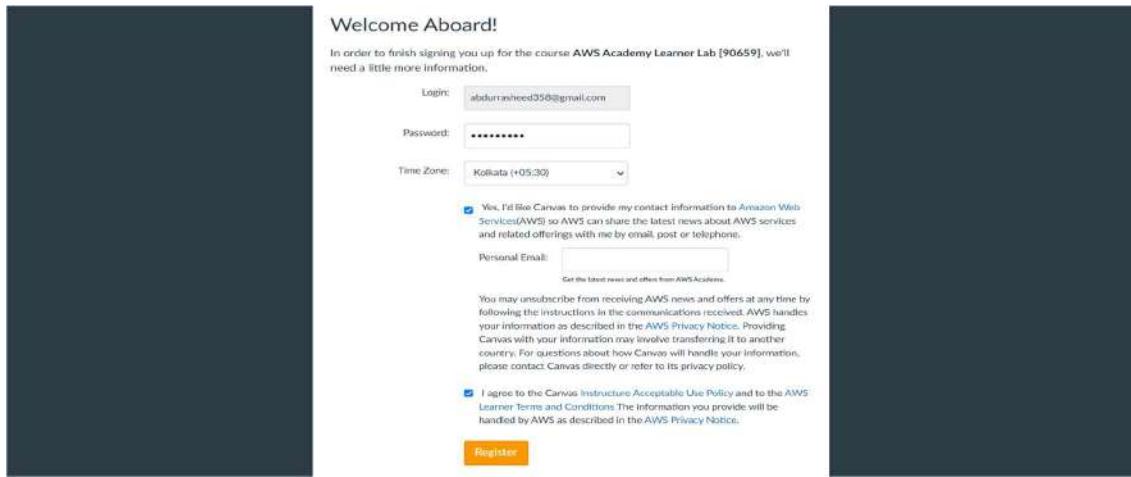
Go to your Email and Check for AWS Academy course invitation.



Click on Get Started.



Click on Create My Account.



- Provide a password.
- Select time zone(any Indian time zone), Check all the boxes.
- Scroll down and Click on Register.

The screenshot shows the main course page for 'AWS Academy Learner Lab [90659]'. The left sidebar has links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area features a banner with a classical building icon and the text 'aws academy'. Below the banner, it says: 'AWS Academy Learner Lab provides a long-running sandbox environment for ad hoc exploration of AWS services. Within this class, students will have access to a restricted set of AWS services. Not all AWS documentation walk-through or sample labs that operate in an AWS Production account will work in the Learner Lab environment. You will retain access to the AWS resources set up in this environment for the duration of this course. We limit your budget (\$50USD), so you should exercise caution to prevent charges that will deplete your budget too quickly. If you exceed your budget, you will lose access to your environment and lose all of your work.' It also states: 'Each session lasts for 4 hours by default, although you can extend a session to run longer by pressing the start button to reset your session timer. At the end of each session, any resources you created will persist. However, we automatically shut EC2 instances down. Other resources, such as RDS instances, keep running. Keep in mind that we do not stop some AWS features, so they can still incur charges between sessions. For example, an Elastic Load Balancer will continue to route traffic to your instances even if they are not running.'

This is the main page of AWS.

- On this window click on account.
- Let's see how to logout and login again →CLICK on logout.

Click on Student Login.

Fill the username and password → click on login.

The screenshot shows the AWS Academy dashboard. On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area displays course details: "AWS Academy Learner Lab [90659]" and "ALLv2EN-US-LTI13-90659". A purple placeholder image is present where course content would normally be. On the right, there are sections for "To Do" (Nothing for now), "Recent Feedback" (Nothing for now), and a "View Grades" button. At the bottom, there are links for Privacy Policy, Cookie Notice, Acceptable Use Policy, Facebook, and X.com.

Click on the AWS academy learner lab.

The screenshot shows the "AWS Academy Learner Lab [90659]" page. The sidebar on the left includes Home, Modules, Discussions, Grades, and Lucid (Whiteboard) options. The main content area features a large graphic of a classical building under a cloud, with the AWS Academy logo at the bottom right. Below the graphic, text explains the purpose of the Learner Lab, mentioning it's a sandbox environment for exploring AWS services. It notes that while some services like RDS are kept running, others like EC2 are stopped after each session. It also cautions against deleting resources like ELB or NAT. A note at the bottom states that session times can be extended by pressing the start button. To the right, there are buttons for View Course Stream, View Course Calendar, and View Course Notifications. The "To Do" section indicates "Nothing for now".

Click on Modules.

The screenshot shows the "Course Modules: AWS Academy Learner Lab [90659]" page. The sidebar on the left lists Home, Modules, Discussions, Grades, and Lucid (Whiteboard). The main content area shows a module titled "Learn how to effectively use the AWS Academy Learner Lab" with a "Module Knowledge Check" (300 pts, Score at least 70.0). Below this, there are two buttons: "AWS Academy Learner Lab" and "Launch AWS Academy Learner Lab". Further down, there's a section for "AWS Academy Learner Lab Resources" containing items like "Demo - How to Access Learner Lab", "Demo - General Troubleshooting Tips", "Demo - How to Launch Services through AWS Console", and "Learner Lab Activity - Amazon Q Developer". At the bottom, there's a "Feedback Survey" section.

Scroll down and click on AWS Academy Learner Lab.

Please read the terms and conditions shown below and click on the "I agree" button at the bottom of this page to continue.

Terms and Conditions

Welcome to the Vocareum, Inc. ("Vocareum") website located at www.vocareum.com (the "Site"). Please read these Terms of Service (the "Terms") and our Privacy Policy (<http://www.vocareum.com/privacy-policy/>) carefully because they govern your use of our Site and our web-based education and learning platform. To make these Terms easier to read, the Site and our platform are collectively called the "Services".

1. Agreement To Terms

By using our Services, you agree to be bound by these Terms. If you don't agree to these Terms, do not use the Services. If you are accessing and using the Services on behalf of an educational institution (such as your employer or the educational institution in which you are enrolled) or other legal entity, you represent and warrant that you have the authority to bind that educational institution or other legal entity to these Terms. In that case, "you" and "your" will refer to that educational institution or other legal entity.

2. Changes to Terms or Services

Scroll down after reading the Terms & Conditions.

these Terms, without such consent, will be null. Vocareum may freely assign or transfer these Terms without restriction. Subject to the foregoing, these Terms will bind and inure to the benefit of the parties, their successors and permitted assigns.

Any notices or other communications provided by Vocareum under these Terms, including those regarding modifications to these Terms, will be given (i) via email, or (ii) by posting to the Services. For notices made by e-mail, the date of receipt will be deemed the date on which such notice is transmitted.

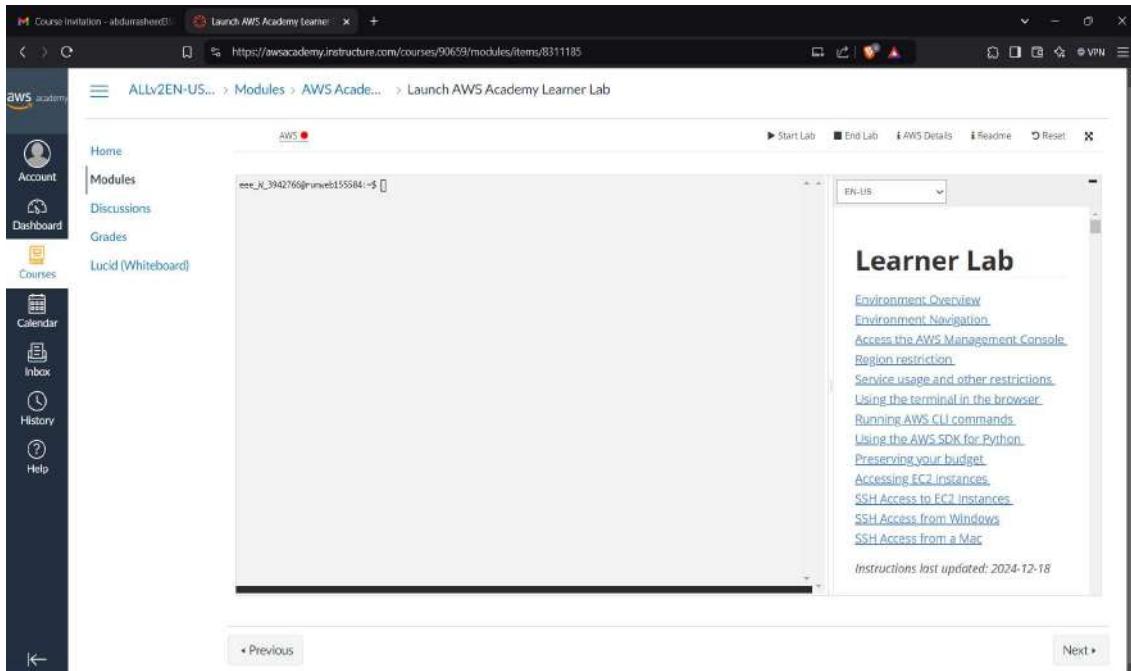
Vocareum's failure to enforce any right or provision of these Terms will not be considered a waiver of such right or provision. The waiver of any such right or provision will be effective only if it is written and signed by a duly authorized representative of Vocareum. Except as expressly set forth in these Terms, the exercise by either party of any of its remedies under these Terms will be without prejudice to its other remedies under these terms or otherwise.

Contact Information

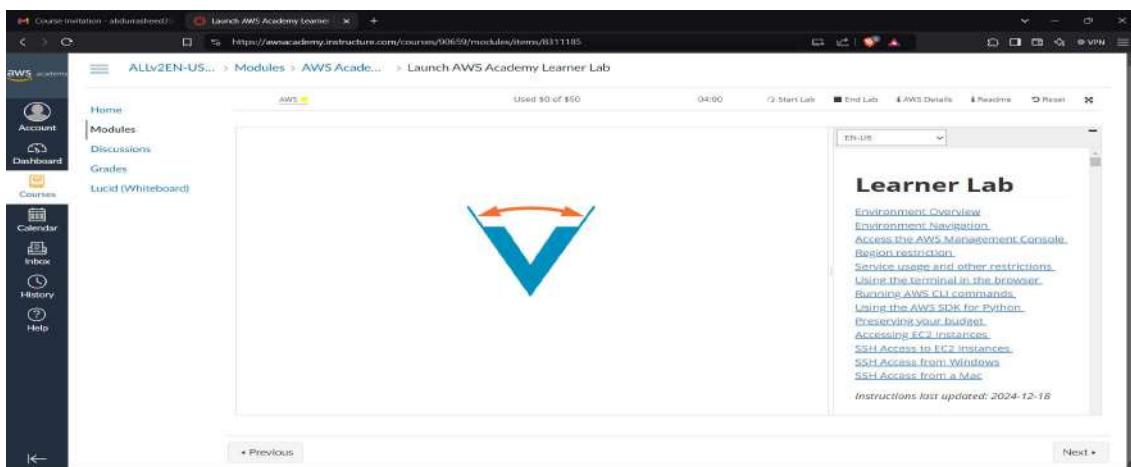
If you have any questions about these Terms or the Services, please contact Vocareum at: info@vocareum.com

I Agree

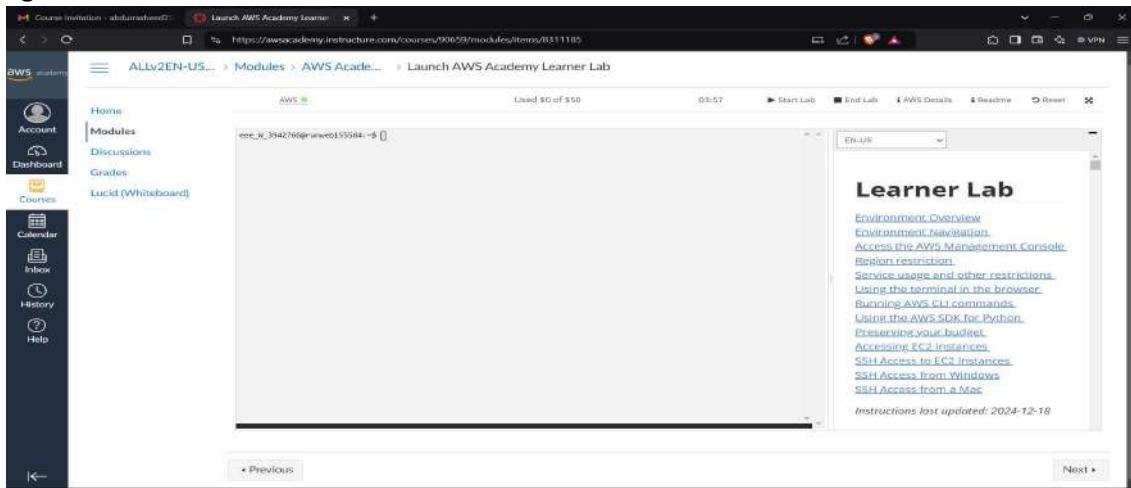
Scroll down and click on I Agree.



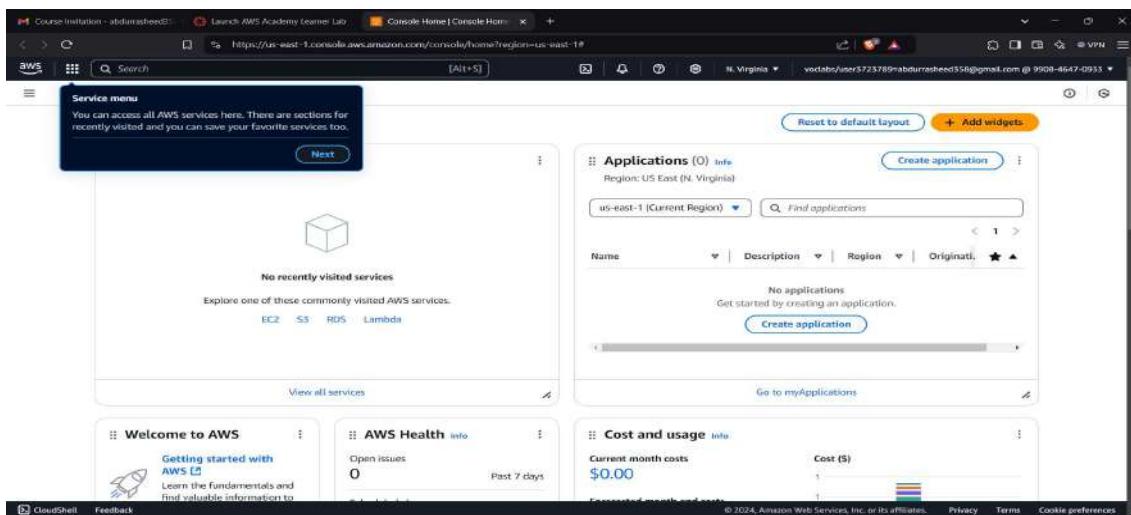
- If there is red dot beside AWS it means the sandbox (AWS Learning Environment) is in stop state, click on Start Lab.
- Once you Start the Lab, it will last for 4hrs and if you want to extend then comeback and start again to reset the timer.



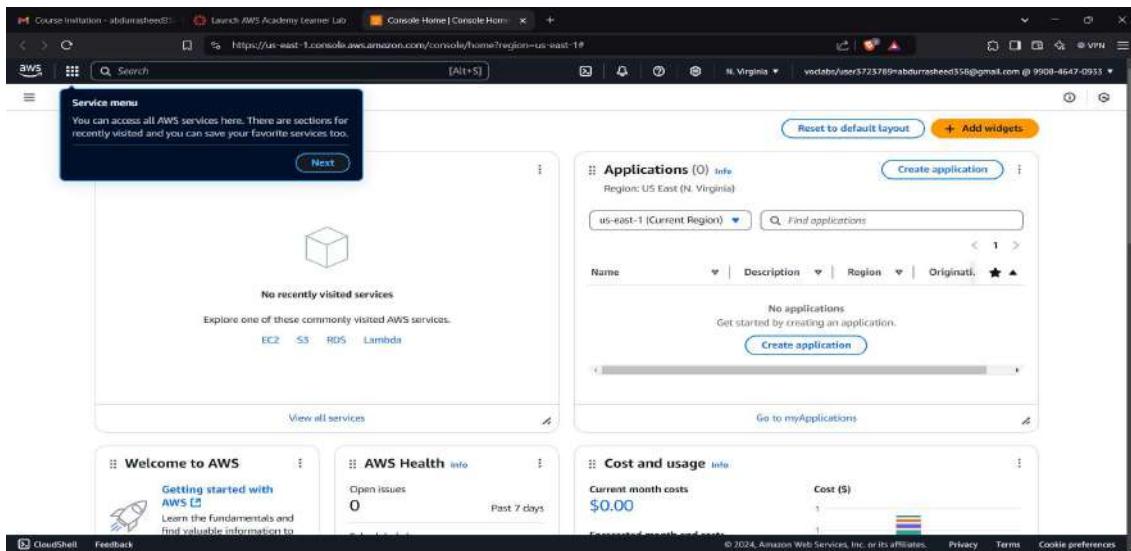
You are given free 50\$, if you exceed you will not be able to access AWS with this email id again.



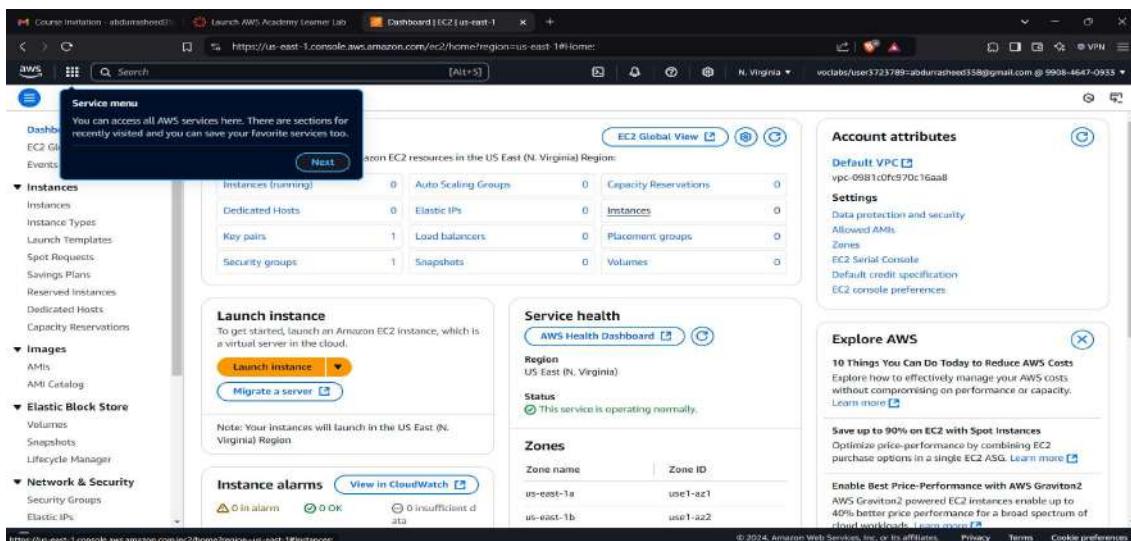
Once the AWS turns green, click on it.



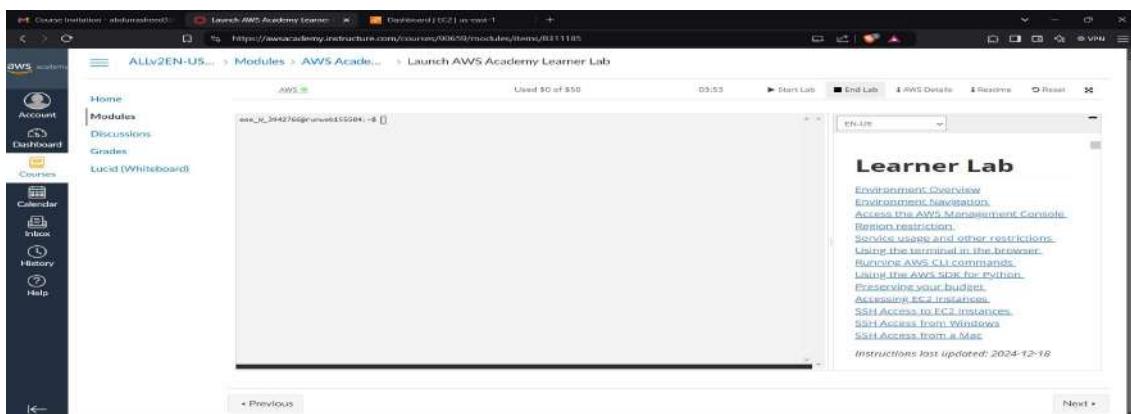
You will land to the AWS dashboard.



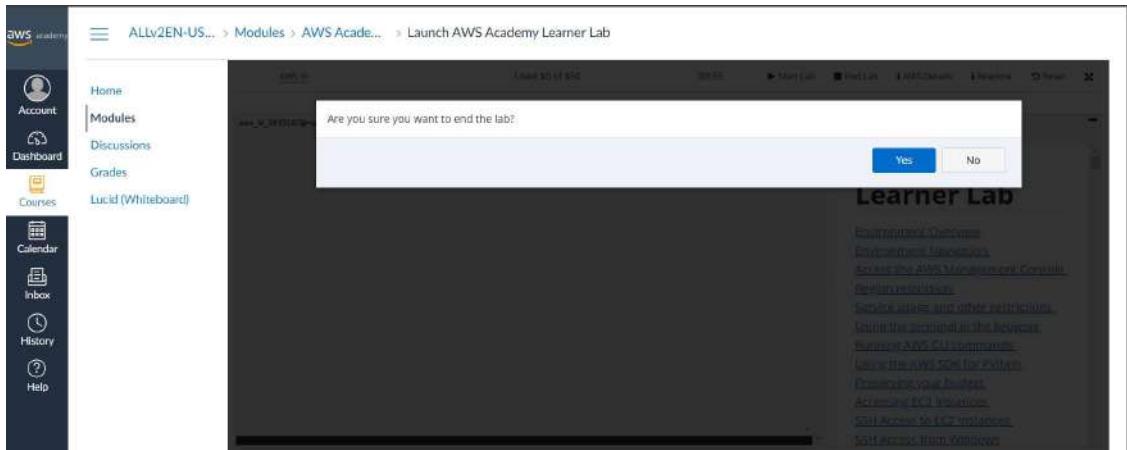
Click on EC2 in Recent visited box.



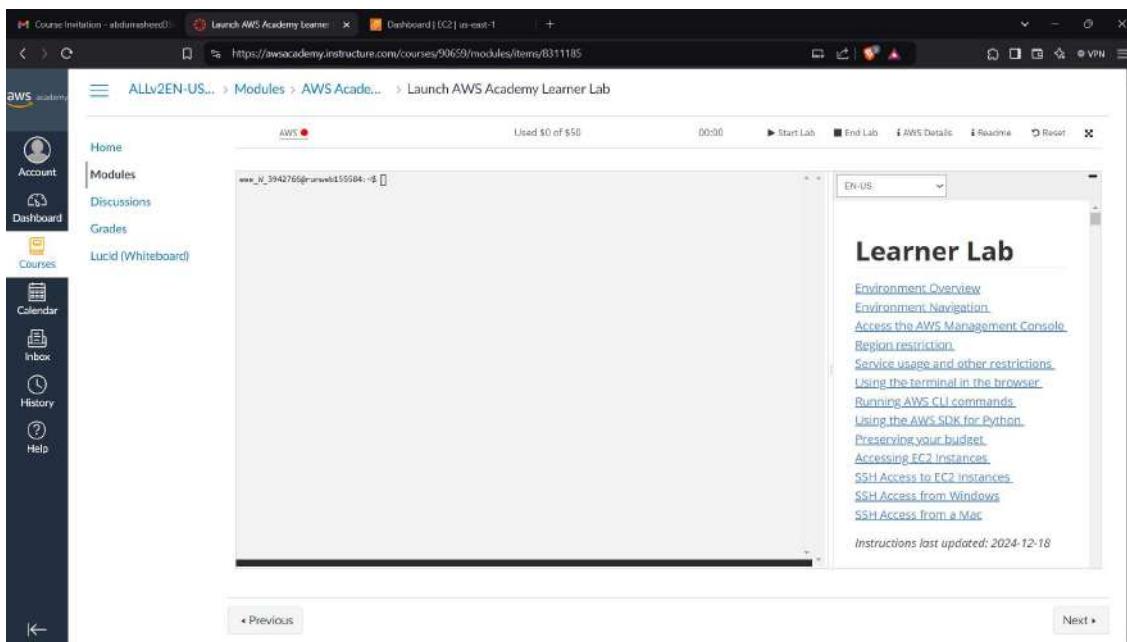
We can start the exercises from here.



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



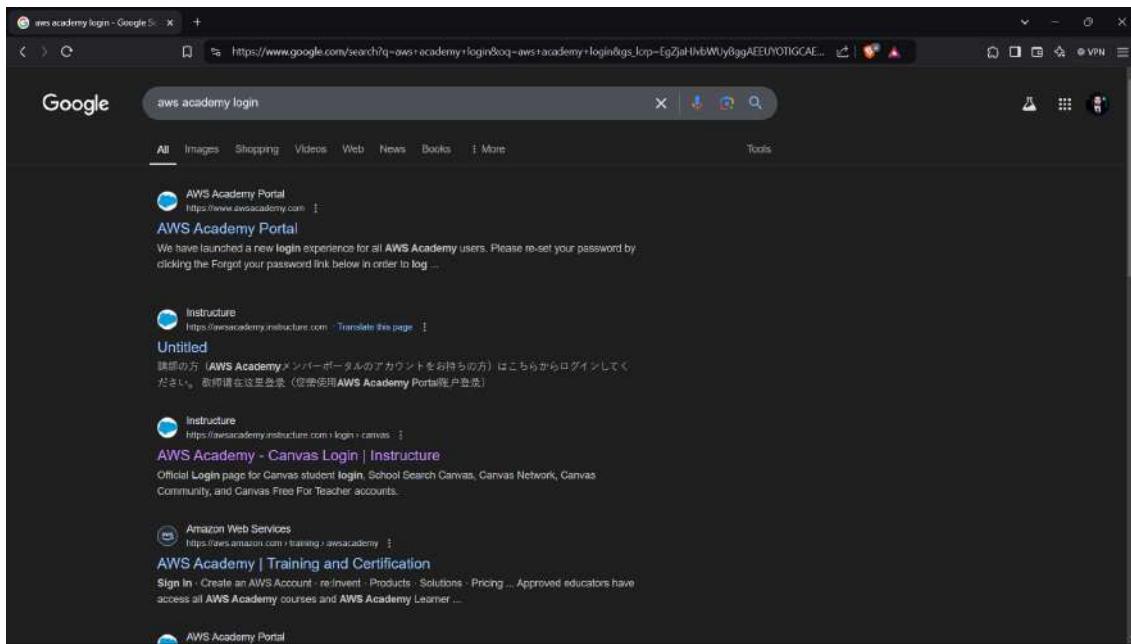
Click on Yes to end the lab.



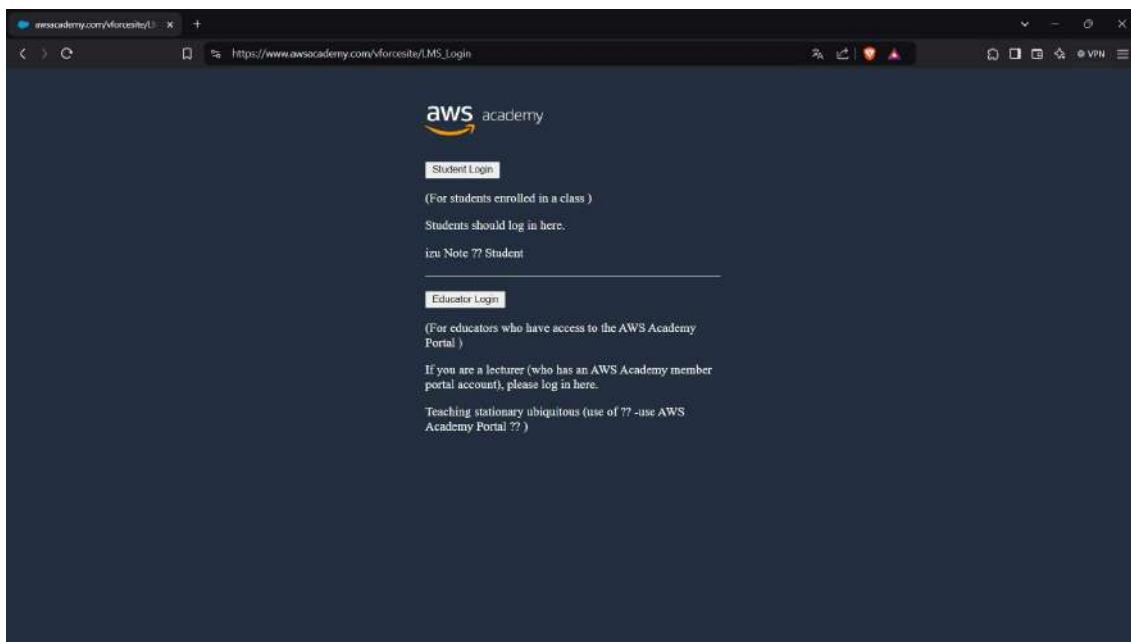
Once you get the red color dot beside AWS , it indicates stopped.

8 B. PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE

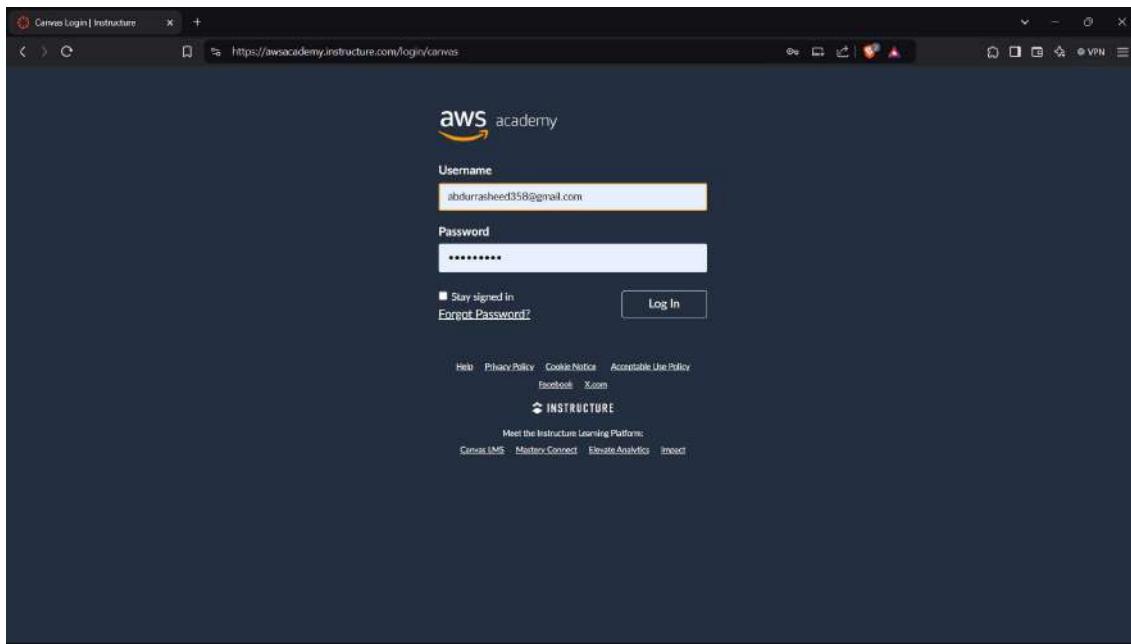
Search for aws Academy Login.



Check for Untitled.



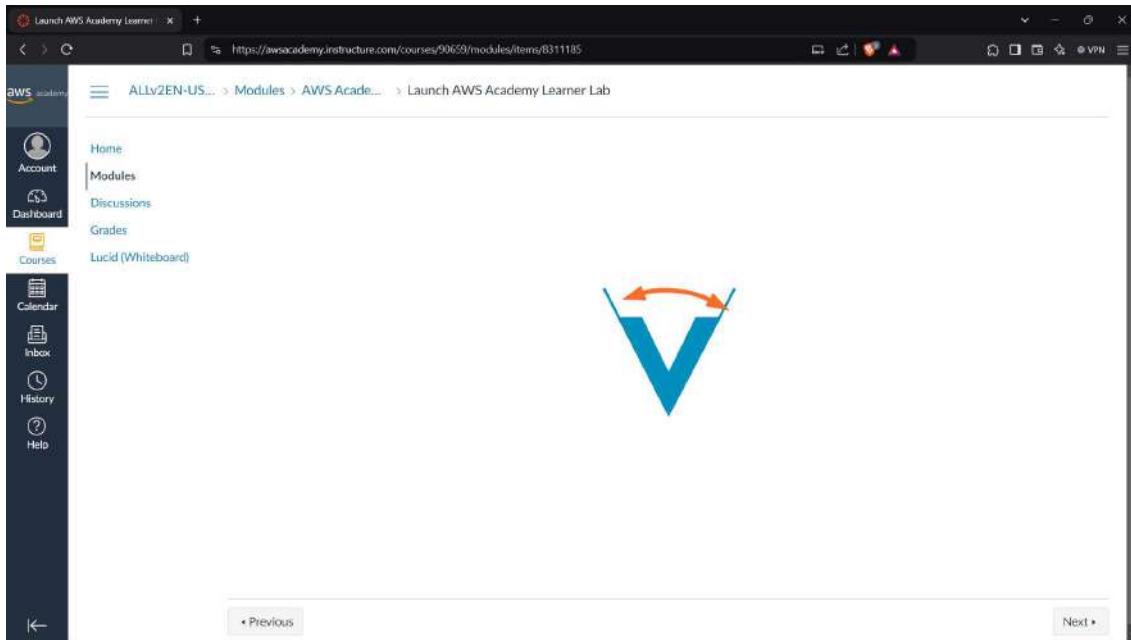
Click on Student Login.



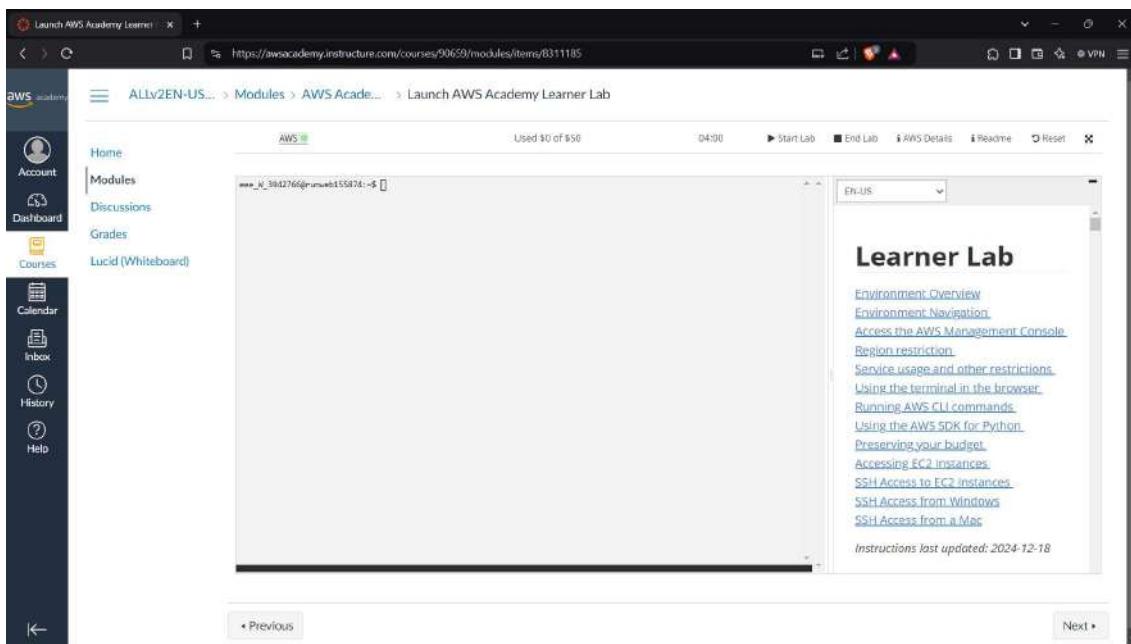
Enter your details and login.

A screenshot of a web browser showing the AWS Academy course modules page. The title bar says "Course Modules: AWS Academy". The URL in the address bar is https://awsacademy.instructure.com/courses/90659/modules. On the left is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area shows course modules. The first module is "AWS Academy Learner Lab Compliance and Security", which includes a "Learn how to effectively use the AWS Academy Learner Lab" section and a "Module Knowledge Check" (100 pts, Score at least 70.0). The second module is "AWS Academy Learner Lab", which has a single item "Launch AWS Academy Learner Lab". The third module is "AWS Academy Learner Lab Resources", which includes items like "Demo - How to Access Learner Lab", "Demo - General Troubleshooting Tips", "Demo - How to Launch Services through AWS Console", and "Learner Lab Activity - Amazon Q Developer". A "Complete All Items" button is visible in the top right of the first module's box.

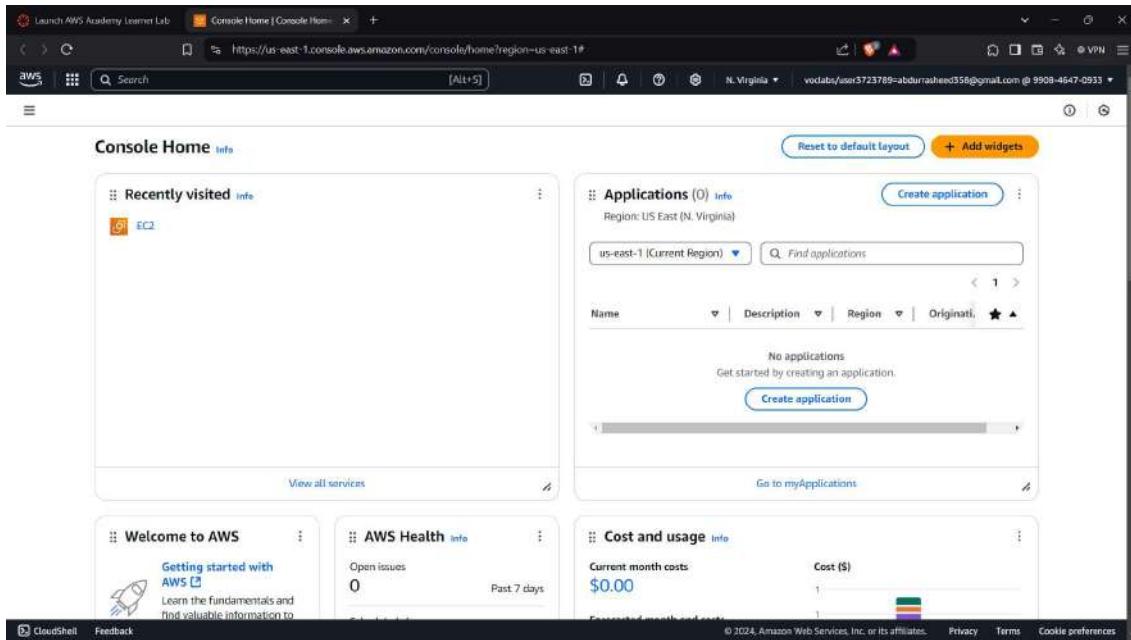
Click on Launch AWS Academy Learner Lab.



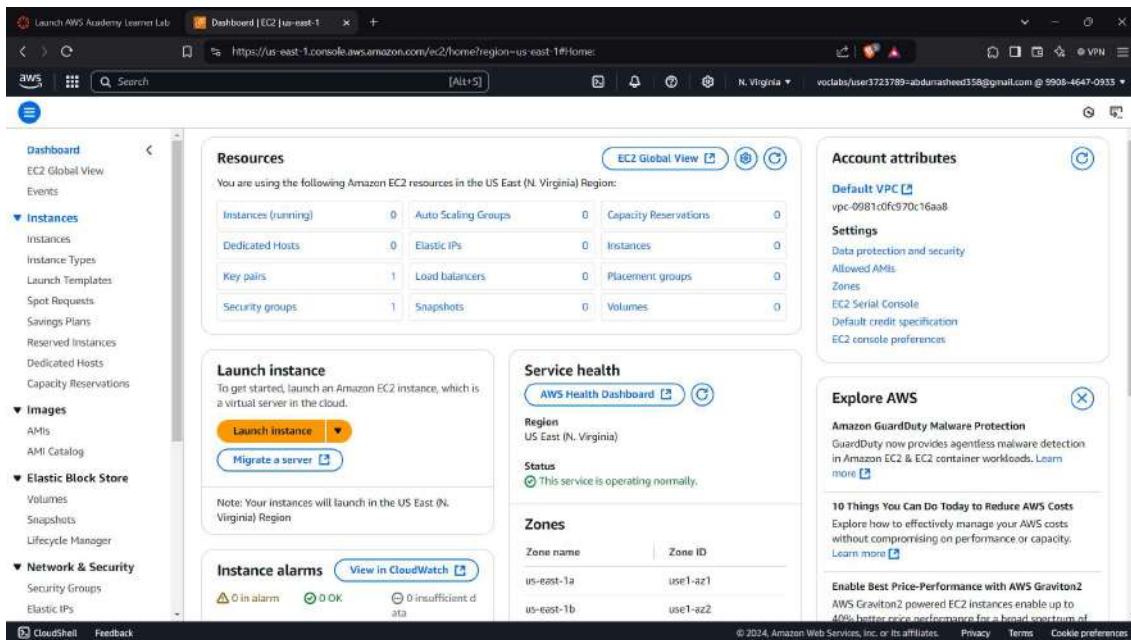
Click on Start Lab.



After the red dot turns to green, click on it.



Click on EC2.



Click Launch Instance.

Name and tags

Name: MyWebAWS

Application and OS Images (Amazon Machine Image)

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

Quick Start

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

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI (ami-01816d0f07b1129dc7c (64-bit (x86), uefi-preferred) / ami-02d4fe51c0559a0e (64-bit (Arm), uefi))

Free tier eligible

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more

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 unavailable instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of

Cancel, Launch instance, Preview code

Name: Enter a name like "MyWebAWS" to identify your server.

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type (ami-0e2c8caa4b6378db8c (64-bit (x86), uefi) / ami-0952fb346e8d448 (64-bit (Arm), uefi))

Free tier eligible

Description

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

Image

64-bit (x86) (selected), 64-bit (Arm), 64-bit (x86)

AMI ID: ami-0e2c8caa4b6378db8c

Username: ubuntu

Verified provider

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

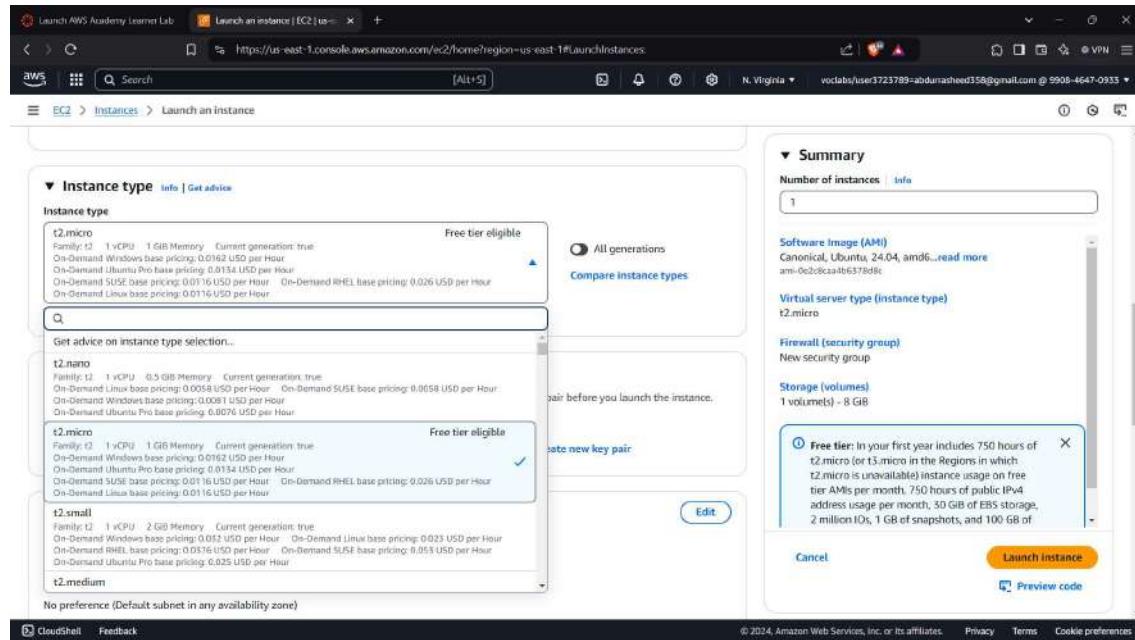
Storage (volumes): 1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Regions in which t2.micro is unavailable instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of

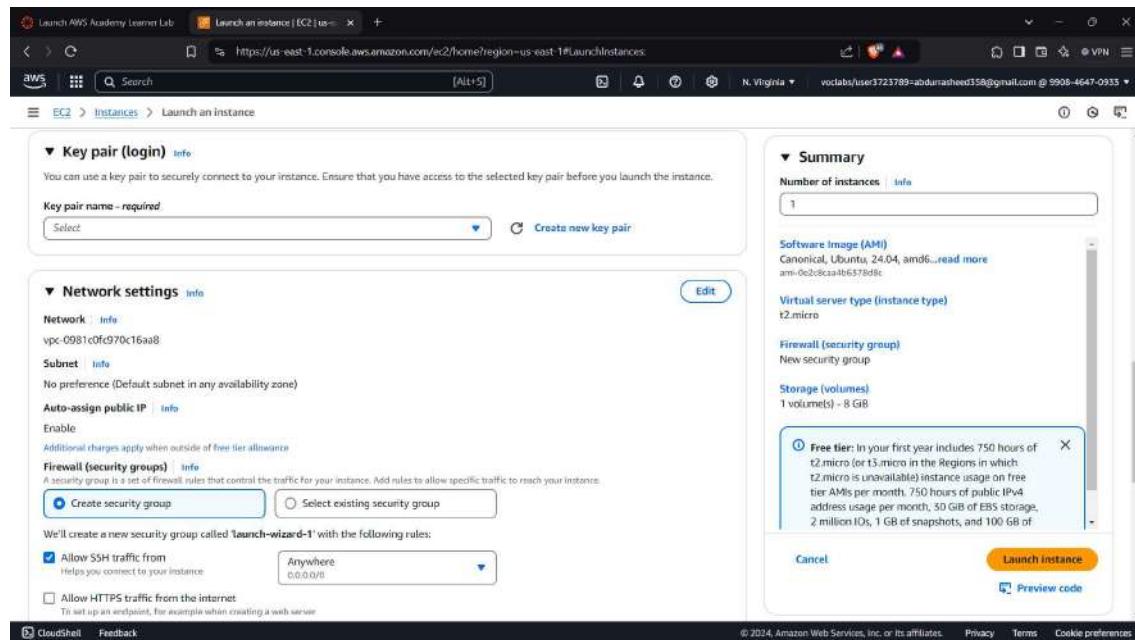
Cancel, Launch instance, Preview code

Application and OS: Choose Ubuntu (Free Tier Eligible).

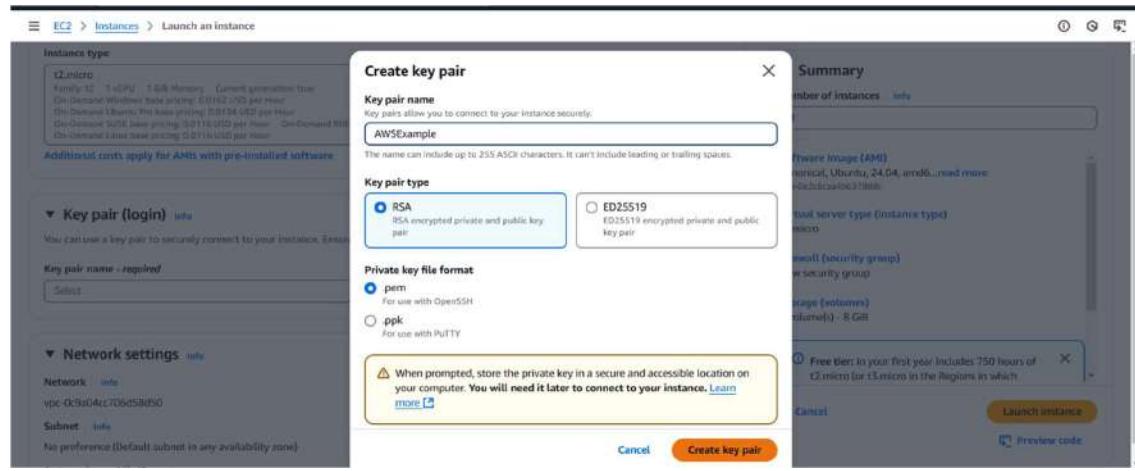
Make sure the architecture is 64-bit(x86).



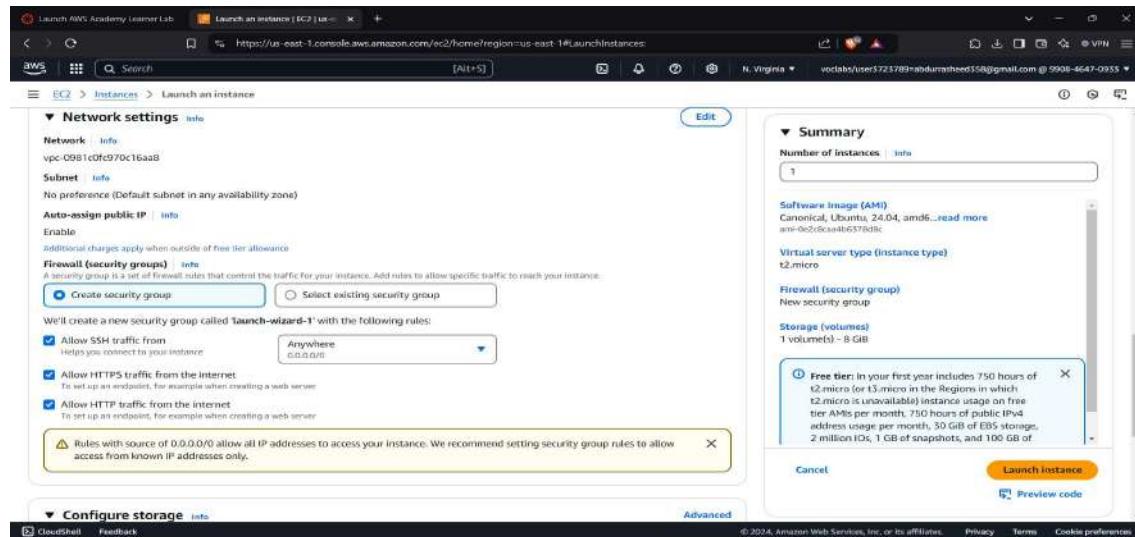
Instance Type: Select t2.micro (1 CPU, 1 GB RAM).



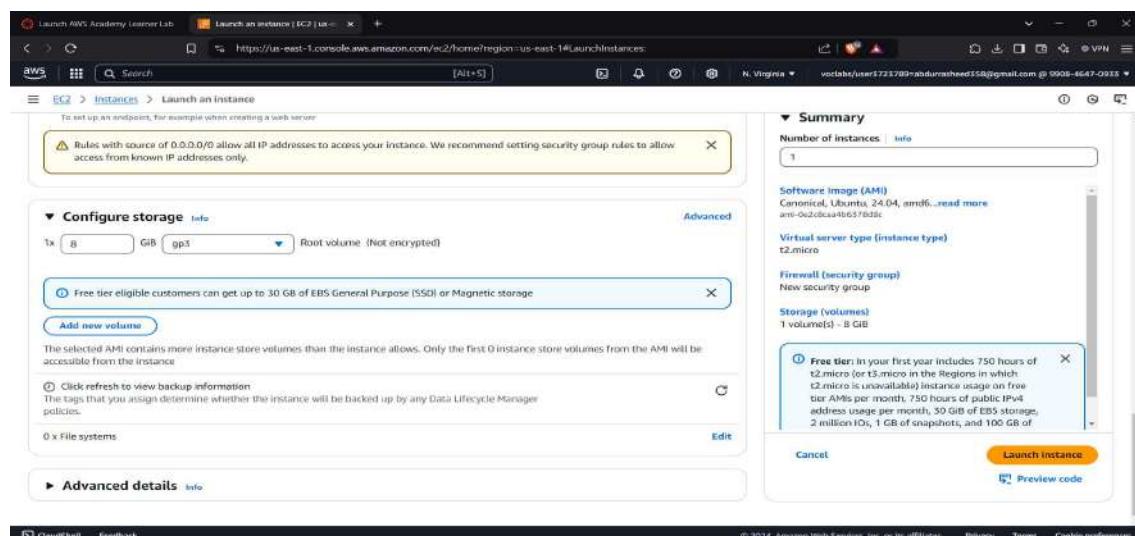
Key Pair: Create a new key pair.



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.



Storage: Use the default 8 GB.

Click Launch Instance.

The screenshot shows the AWS Lambda console with a success message: "Successfully initiated launch of instance (i-03ad65c4a3b94b3b1)". Below this, there's a "Next Steps" section with several options:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the instance will collect more detailed metrics. Includes a "Learn more" link.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create a new RDS database" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs. Includes a "Learn more" link.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "View alarms" button.

At the bottom, there are links for CloudShell, Feedback, and a copyright notice: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

The instance is initiated successfully.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation menu with "Instances" selected. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
MyWebAWS	i-03ad65c4a3b94b3b1	Running	t2.micro	Initializing		us-east-1c	ec2-23-23-

Below the table, a modal window titled "Select an instance" is open, showing the same instance details.

Click on Instances->wait until the status changes to "Running."

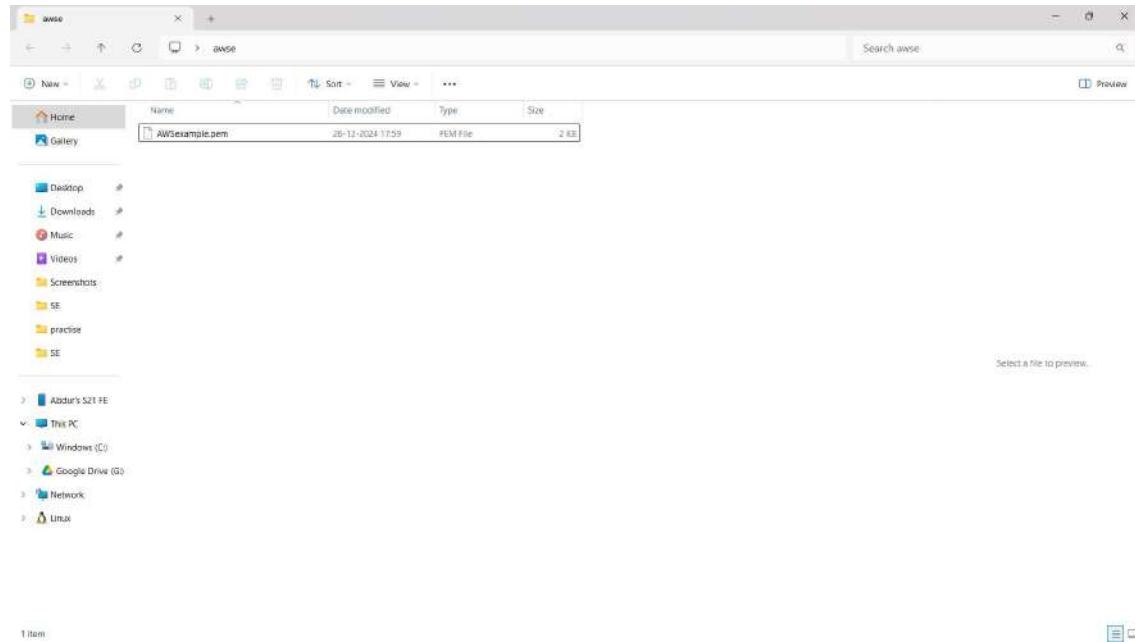
The screenshot shows the "Connect to instance" interface for the EC2 instance "MyWebAWS". It includes tabs for "EC2 Instance Connect", "Session Manager", "SSH client" (which is selected), and "EC2 serial console".

The "SSH client" tab shows the following information:

- Instance ID: i-03ad65c4a3b94b3b1 (MyWebAWS)
- SSH command copied: ssh -i "AWSexample.pem" ubuntu@ec2-23-23-71-60.compute-1.amazonaws.com
- 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.

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

`ssh -i "AWSexample.pem" ubuntu@ec2-23-23-71-60.compute-1.amazonaws.com`



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

Copy the path of the folder.

Open PowerShell (Windows) or Terminal (Mac/Linux) on your computer->Run As Administrator.

```
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\abdur\OneDrive\Desktop\awse
PS C:\Users\abdur\OneDrive\Desktop\awse>
```

Navigate to the folder where your .pem file is saved using the cd command.

```
ubuntu@ip-172-31-29-33:~$ ssh -i C:/Users/abdur/OneDrive/Desktop/awse ubuntu@ec2-23-23-71-60.compute-1.amazonaws.com
The authenticity of host 'ec2-23-23-71-60.compute-1.amazonaws.com (23.23.71.60)' can't be established.
ED25519 key fingerprint is SHA256:ec2aBbEvaWicDr-Vwwu1YkFSyrcDlVzmcfyPvNMo3g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-23-23-71-60.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1918-aws x86_64)

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

System information as of Thu Dec 26 12:40:36 UTC 2024

System load: 0.0 Processes: 105
Usage on /: 7.7% of 6.73GB Users logged in: 0
Memory usage: 28%
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-29-33:~$
```

Paste the SSH command and press Enter.

Type "yes" if prompted.

Update the system to ensure all software is up to date:

```
sudo apt update
```

```
[ubuntu@ip-172-31-29-33:~]$ Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [110 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3844 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.7 kB]
ubuntu@ip-172-31-29-33:~$ Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.7 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
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 [52.0 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 [560 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 [212 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [12.2 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 [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 31.8 MB in 6s (5454 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-172-31-29-33:~$
```

Check if the updates are done.

```

ubuntu@ip-172-31-29-33:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd 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 containerd 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.
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 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-updates/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 1s (83.6 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.

```

Install Docker to package and run our web application:

`sudo apt-get install docker.io`

```

ubuntu@ip-172-31-29-33:~$ 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-172-31-29-33:~$ 

```

Install Git to manage and download code:

`sudo apt install git`

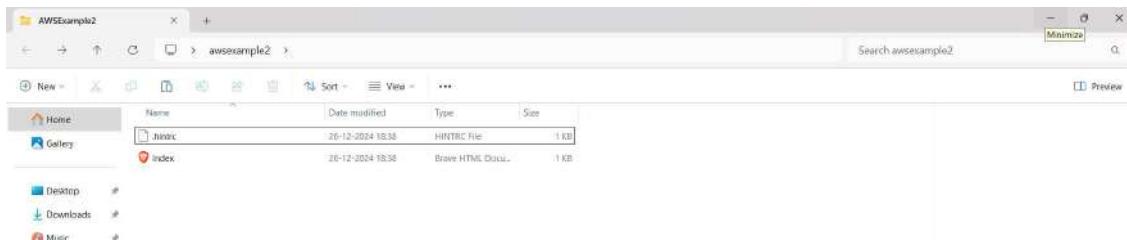
```

ubuntu@ip-172-31-29-33:~$ sudo apt install nano
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-172-31-29-33:~$ 

```

Install Nano for editing files directly on the server:

`sudo apt install nano`



Create Your Web Application

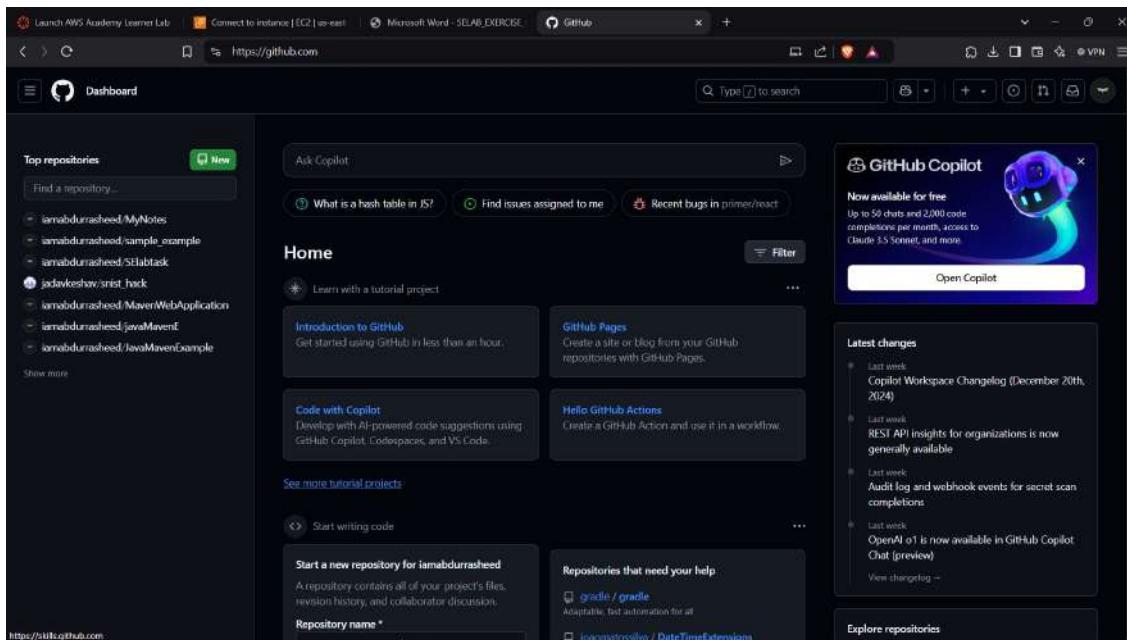
In this step, we will build a simple web page and upload it to GitHub.

Create a folder in your desktop.

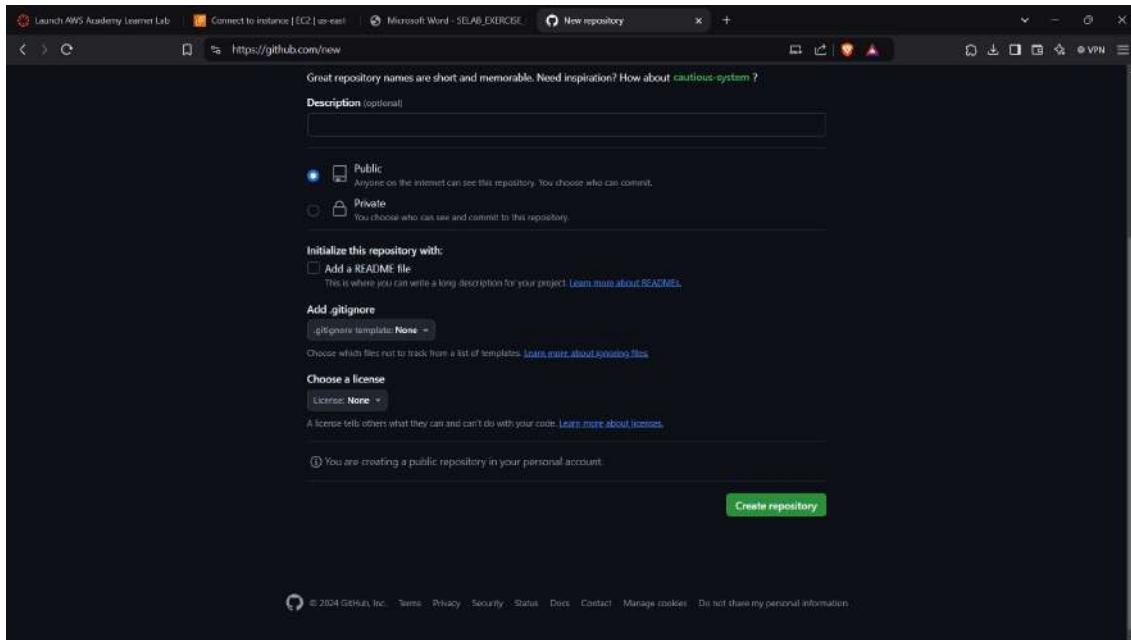


On your computer, create a file named index.html and add the following content:

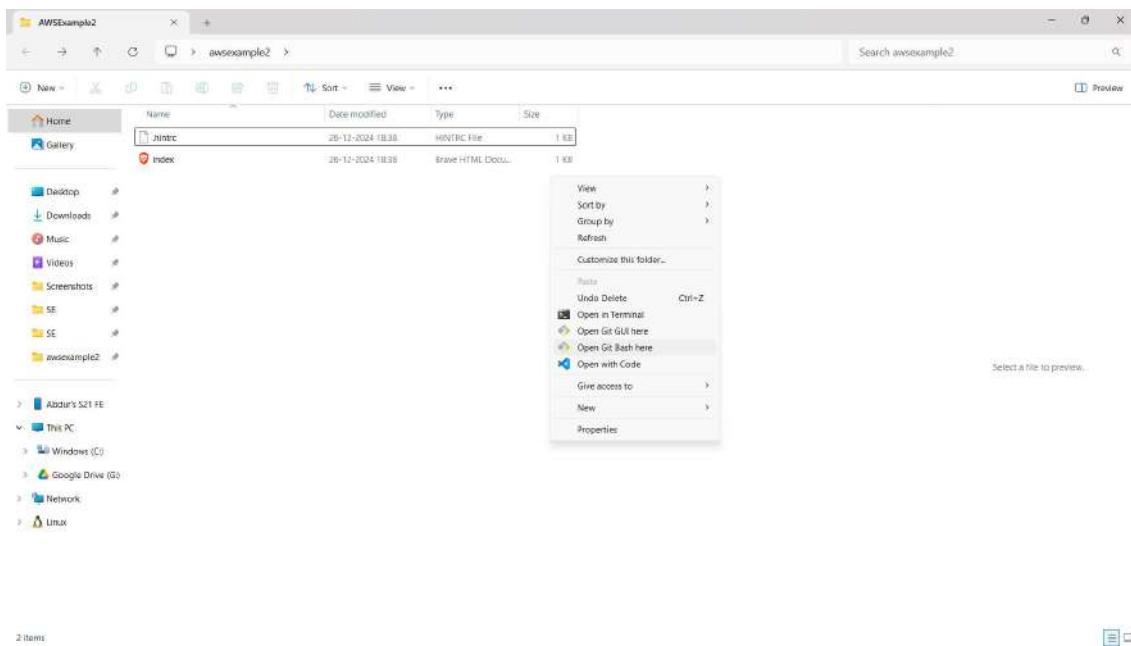
Hello from AWS!



Open your github account -> Create new repository.



Give a name to your repository and click on Create Repository.



Right click on folder -> Open Git Bash Here.

```
MINGW64:/c/Users/abdur/OneDrive/Desktop/awsexample2
$ ls
index.html

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2
$ git init
Initialized empty Git repository in C:/Users/abdur/OneDrive/Desktop/awsexample2/
.git/

IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (master)
$ |
```

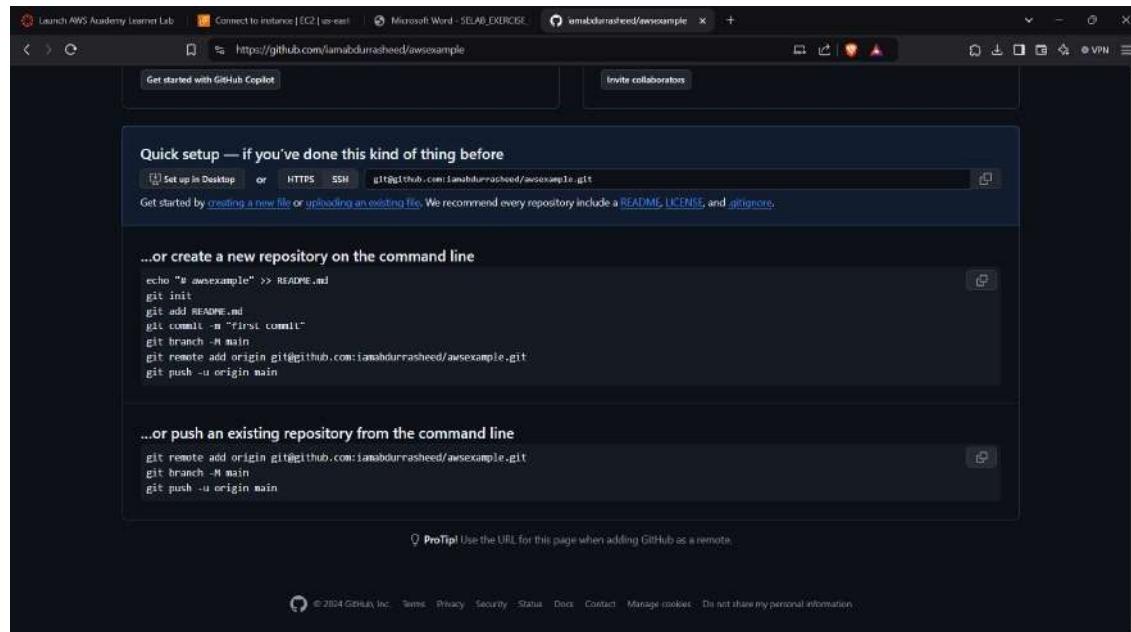
Initialize Git in the file's folder:

git init

```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (master)
$ git add .
warning: in the working copy of '.hintrc', LF will be replaced by CRLF the next time Git touches it
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (master)
$ git commit -m "Created AWS example"
[master (root-commit) 1119b86] created AWS example
 2 files changed, 21 insertions(+)
 create mode 100644 .hintrc
 create mode 100644 index.html
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (master)
$
```

git add . to stage all changes.

git commit -m "Created AWS Example" to commit the changes.



Copy the last three commands.

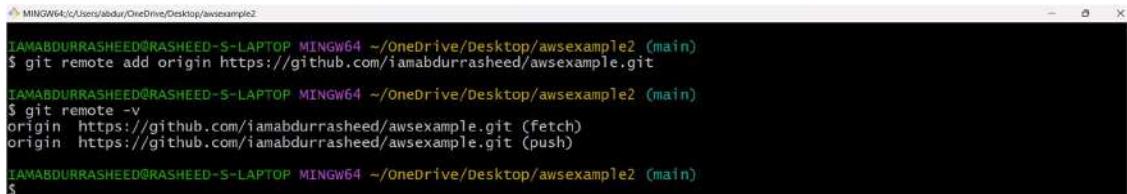
```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (master)
$ git branch -M main
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$
```

git branch -M main to change your branch to main.



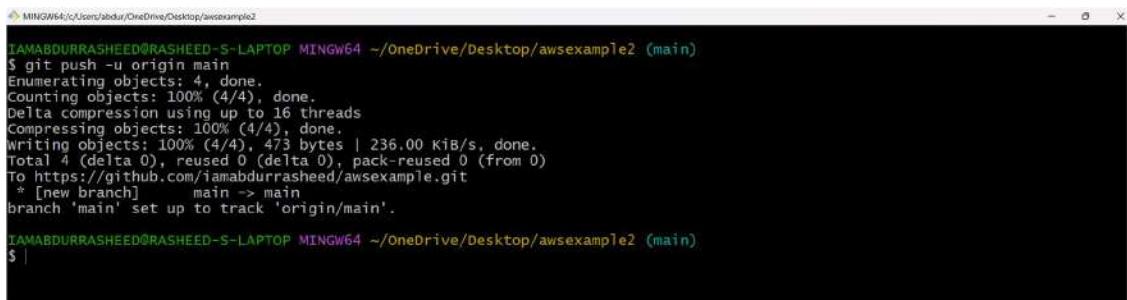
```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$ git remote add origin https://github.com/iamabdurrasheed/awsexample.git
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$
```

git remote add origin “your repo URL”.



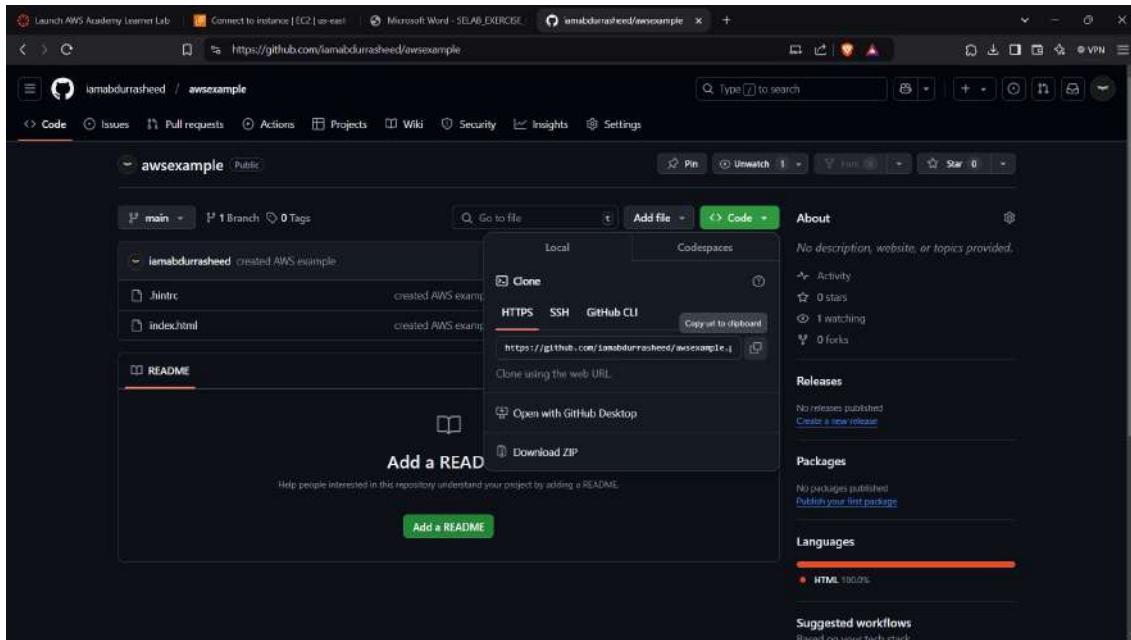
```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$ git remote add origin https://github.com/iamabdurrasheed/awsexample.git
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$ git remote -v
origin https://github.com/iamabdurrasheed/awsexample.git (fetch)
origin https://github.com/iamabdurrasheed/awsexample.git (push)
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$
```

git remote -v shows addresses of your repository.



```
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 473 bytes | 236.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/iamabdurrasheed/awsexample.git
 * [new branch]    main > main
branch 'main' set up to track 'origin/main'.
IAMABDURRASHEED@RASHEED-S-LAPTOP MINGW64 ~/OneDrive/Desktop/awsexample2 (main)
$ |
```

git push -u origin main to push the repo.



Copy the repo URL.

<https://github.com/iamabdurrasheed/awsexample.git>

```
ubuntu@ip-172-31-29-33:~$ sudo git --version
git version 2.43.0
ubuntu@ip-172-31-29-33:~$ sudo docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-29-33:~$ git clone https://github.com/iamabdurrasheed/awsexample.git
Cloning into 'awsexample'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
ubuntu@ip-172-31-29-33:~$
```

On the EC2 instance, clone your GitHub repository:

git clone "your repo URL".

```
ubuntu@ip-172-31-29-33:~$ sudo git --version
git version 2.43.0
ubuntu@ip-172-31-29-33:~$ sudo docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-29-33:~$ git clone https://github.com/iamabdurrasheed/awsexample.git
Cloning into 'awsexample'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
ubuntu@ip-172-31-29-33:~$ ls
awsexample
ubuntu@ip-172-31-29-33:~$
```

Check if repo is cloned or not by typing ls.



```
ubuntu@ip-172-31-29-33:~/awsexample
ubuntu@ip-172-31-29-33:~/awsexample$ cd awsexample
ubuntu@ip-172-31-29-33:~/awsexample$
```

Change your directory to the repo.



```
ubuntu@ip-172-31-29-33:~/awsexample
ubuntu@ip-172-31-29-33:~/awsexample$ ls
index.html
ubuntu@ip-172-31-29-33:~/awsexample$
```

Type ls to check for files.



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

The terminal shows the Dockerfile content:

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

The bottom of the screen displays the nano editor's command-line interface with various keyboard shortcuts.

Add the following content:

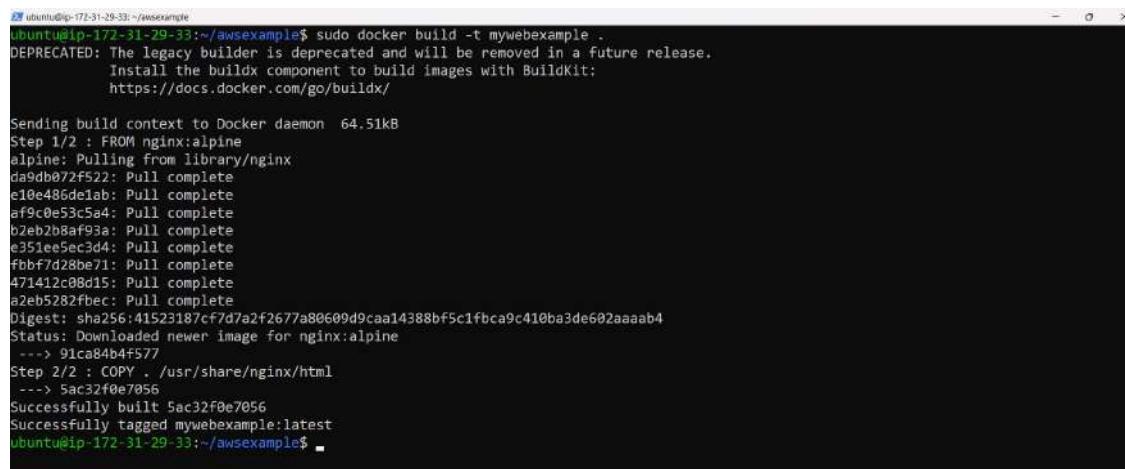
```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

Save the file by pressing Ctrl + O, then Enter, and exit Nano with Ctrl + X.



```
ubuntu@ip-172-31-29-33:~/awsexample$ nano Dockerfile
ubuntu@ip-172-31-29-33:~/awsexample$ ls
Dockerfile index.html
ubuntu@ip-172-31-29-33:~/awsexample$ -
```

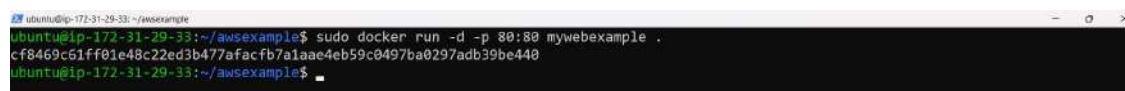
Type ls to check for files.



```
ubuntu@ip-172-31-29-33:~/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 64.51kB
Step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
da9db072f522: Pull complete
e10e486de1ab: Pull complete
af9c0e53c5a4: Pull complete
b2eb208af93a: Pull complete
e351ee5ec3d4: Pull complete
fbfbf7d28be71: Pull complete
471412c08d15: Pull complete
a2eb5282fbec: Pull complete
Digest: sha256:a1523187cf7d7a2f2677a80609d9caa14388bf5c1fbca9c410ba3de602aaaab4
Status: Downloaded newer image for nginx:alpine
--> 91ca84b4f577
Step 2/2 : COPY . /usr/share/nginx/html
--> 5ac32f0e7056
Successfully built 5ac32f0e7056
Successfully tagged mywebexample:latest
ubuntu@ip-172-31-29-33:~/awsexample$ -
```

Build Docker container:

`sudo docker build -t mywebexample .`



```
ubuntu@ip-172-31-29-33:~/awsexample$ sudo docker run -d -p 80:80 mywebexample .
cf8469c61ff01e48c22ed3b477afacfb7a1aae4eb59c0497ba0297adb39be440
ubuntu@ip-172-31-29-33:~/awsexample$ -
```

Run the Docker container to serve the web application:

`sudo docker run -d -p 80:80 mywebexample .`

The screenshot shows the AWS CloudWatch Metrics interface. A metric named 'CPUUtilization' is selected for the instance 'MyWebAWS'. The chart displays the metric value over a period of time, with a red shaded area representing the range of data. The X-axis represents time, and the Y-axis represents the metric value.

Copy the Public IP Address of your EC2 instance from the AWS console.

The screenshot shows a web browser window displaying the EC2 instance's public IP address, 23.23.71.60. The page content is "Hello! from AWS".

Paste it into your browser (e.g., `http://<Public_IP>`).

You'll see your web page with the message "Hello from AWS!" displayed.

```
ubuntu@ip-172-31-29-33:~/awsexample$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
cff7bafceebd      mywebexample       "/docker-entrypoint..."   3 minutes ago     Up 3 minutes      0.0.0.0:80->80/tcp, :::80->80/tcp   determined_no
rthcutt
ubuntu@ip-172-31-29-33:~/awsexample$
```

`sudo docker ps` to display running containers.

Copy the container ID.

```
ubuntu@ip-172-31-29-33:~/awsexample$ sudo docker stop cff7bafceebd
cff7bafceebd
ubuntu@ip-172-31-29-33:~/awsexample$
```

Stop the running Docker container:

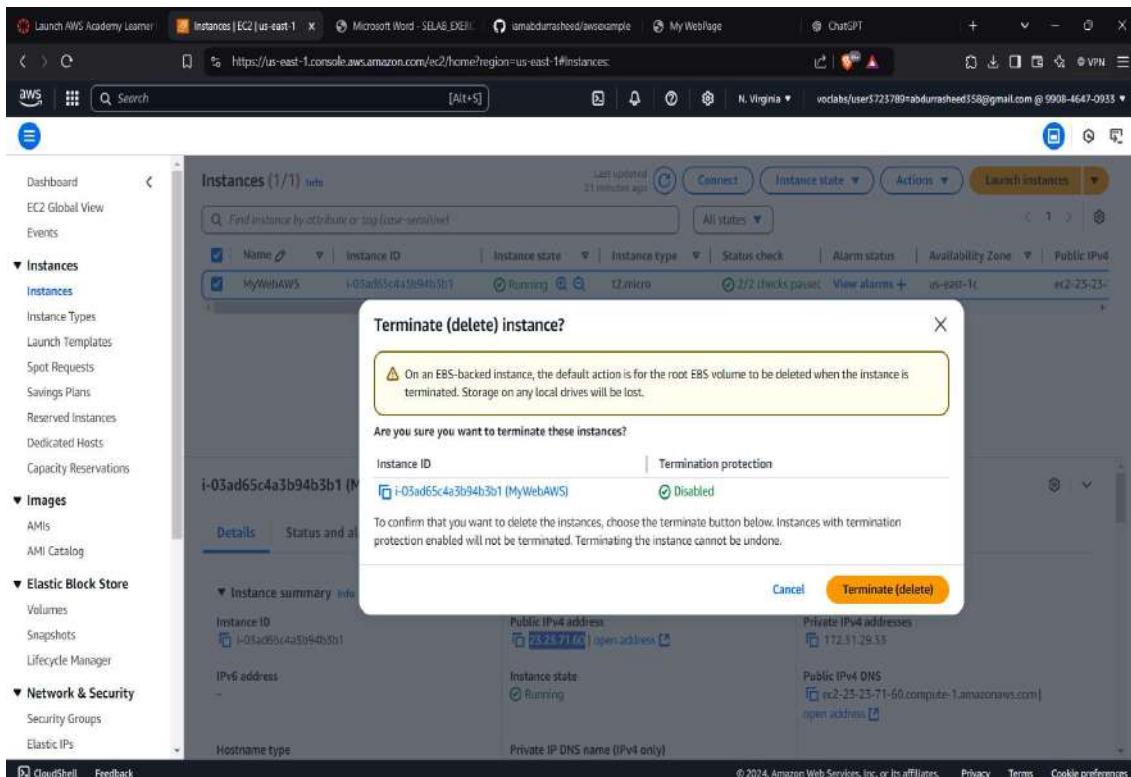
`sudo docker stop "Container ID".`

```
Administrator: Windows PowerShell
ubuntu@ip-172-31-29-33:~/awsexample$ sudo docker stop cff7bafceebd
cff7bafceebd
ubuntu@ip-172-31-29-33:~/awsexample$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-172-31-29-33:~/awsexample$ exit
logout
Connection to ec2-23-23-71-60.compute-1.amazonaws.com closed.
PS C:\Users\abdur\OneDrive\Desktop\aws>
```

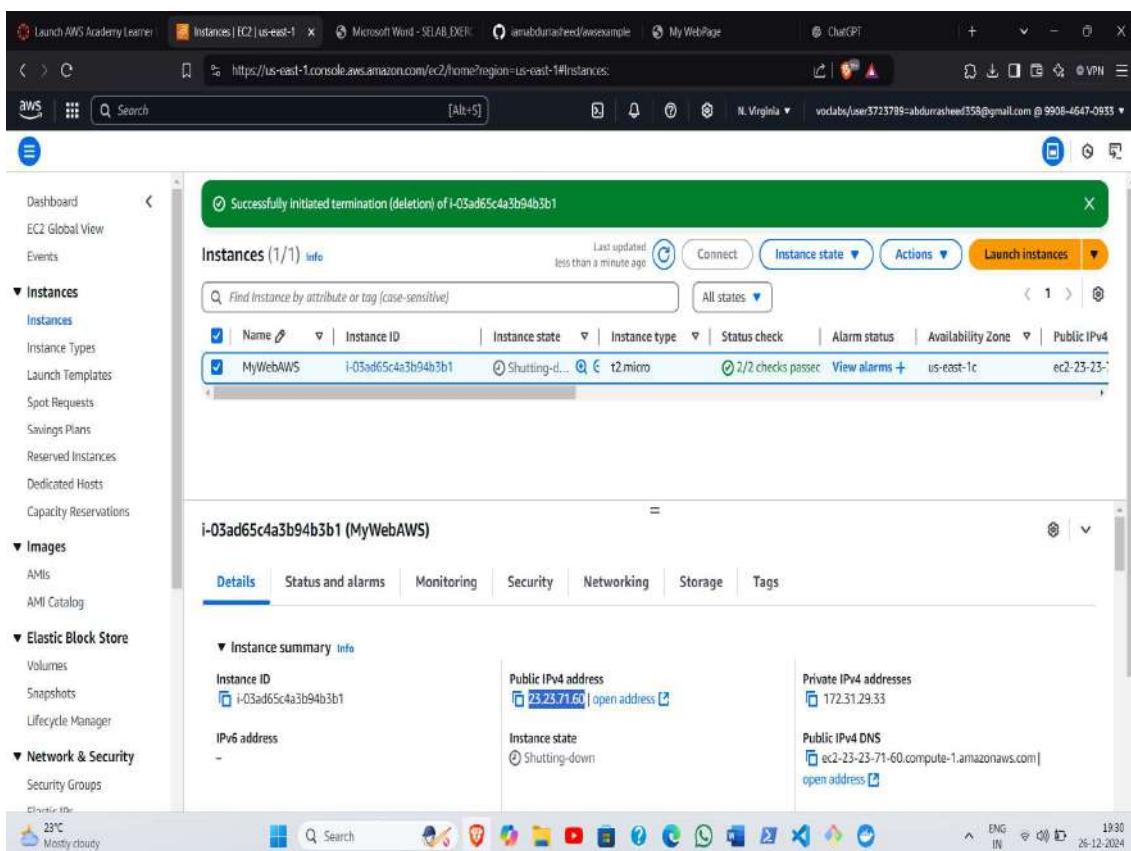
Type exit to logout.

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a navigation pane with options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main area displays a chart titled 'CPU' with a single data series. The chart shows a constant value of 100% across all metrics. Below the chart, there's a table with columns: Metric Name, Value, and Last updated. The table shows a single row with 'CPU' as the metric name, '100%' as the value, and '20 minutes ago' as the last update time.

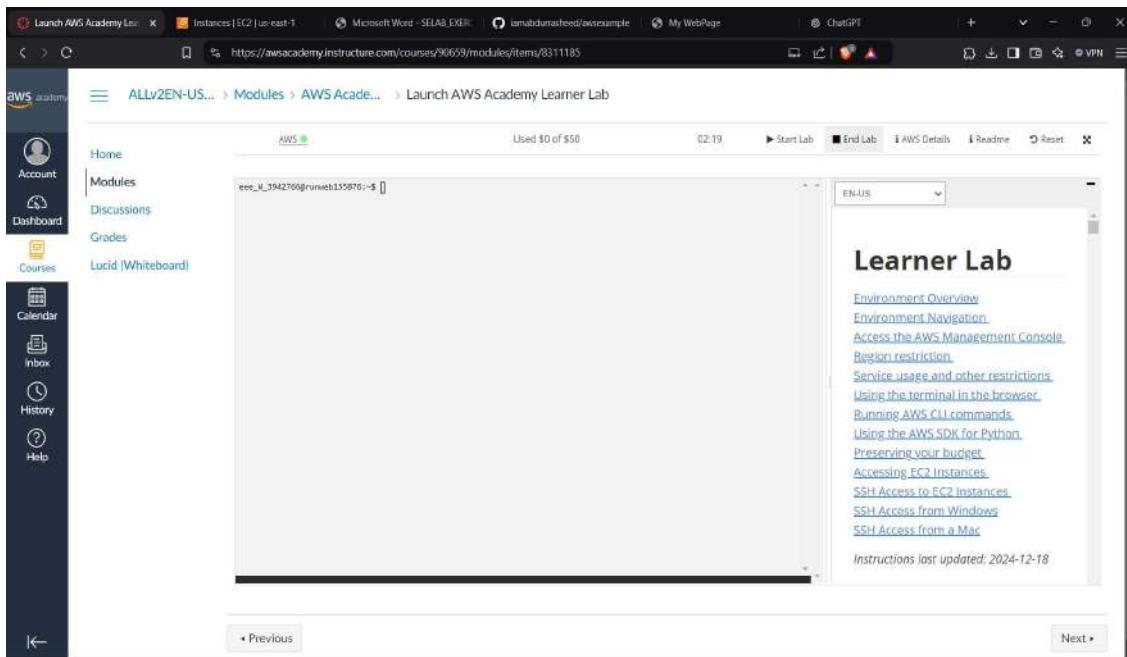
Terminate the EC2 instance in the AWS console by selecting it, clicking Instance State, and choosing Terminate Instance.



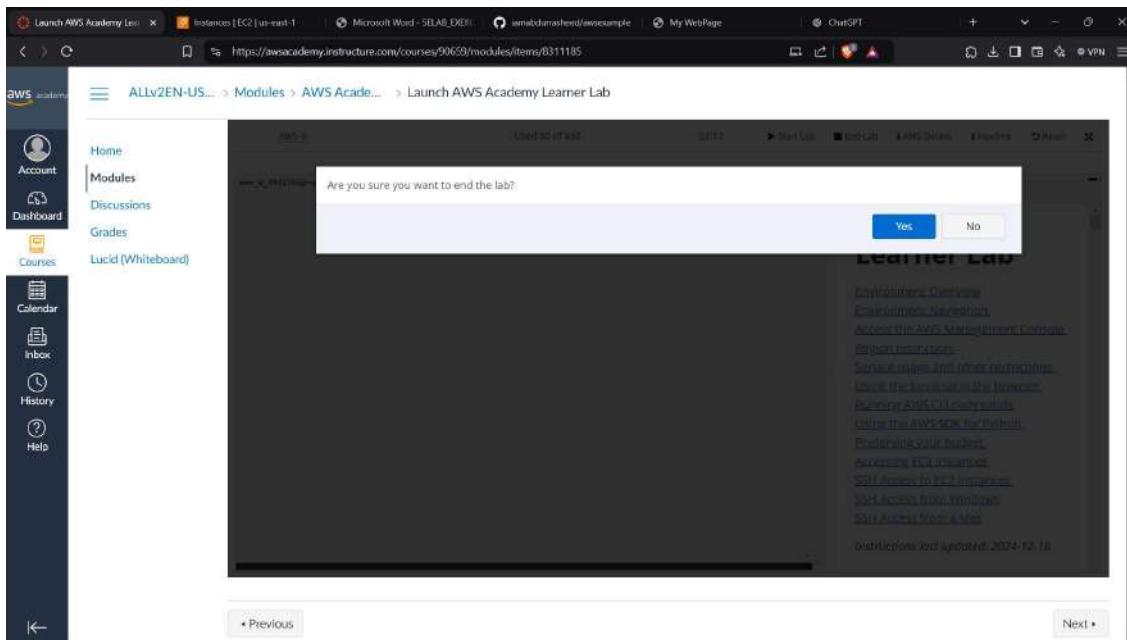
Click on Terminate(delete).



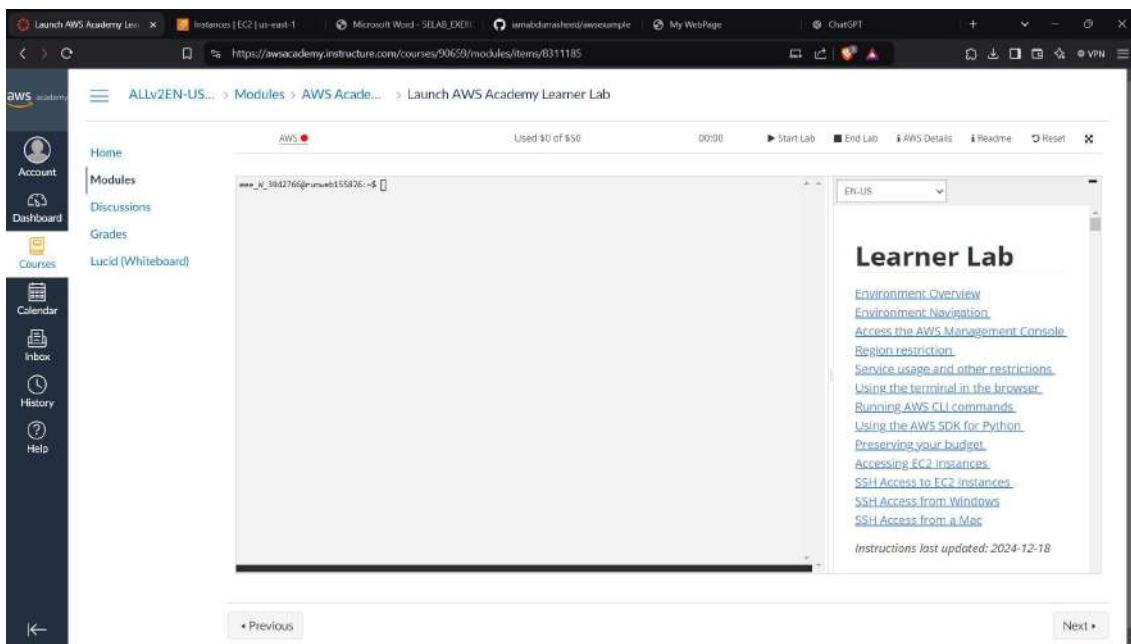
Check if the state is Terminated or not under Instance state.



Click on End Lab.



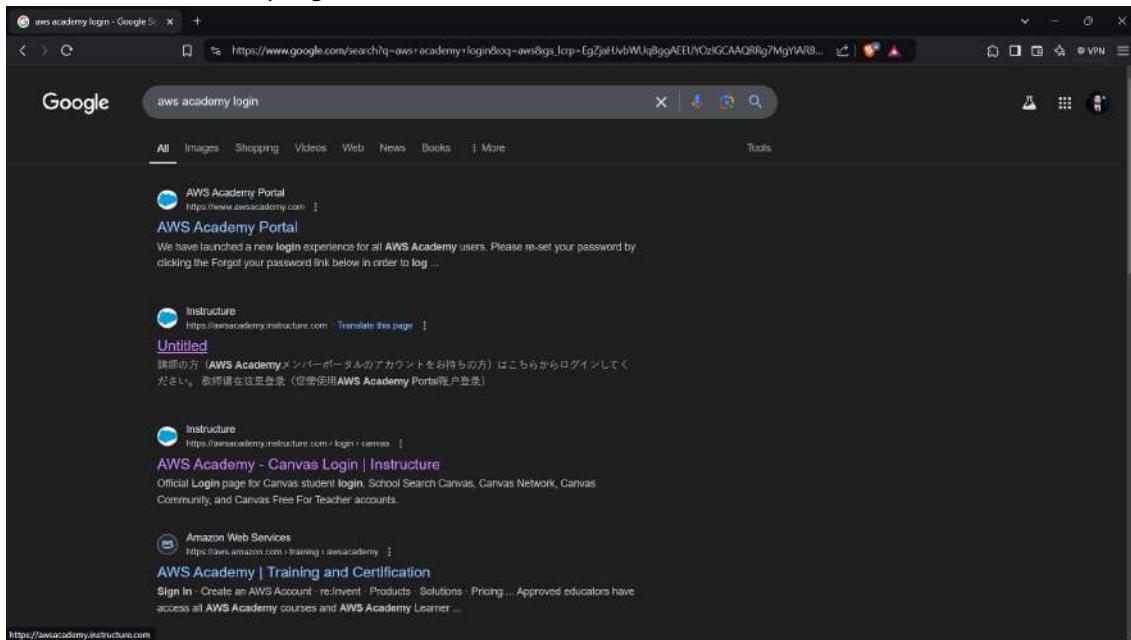
Click on Yes.



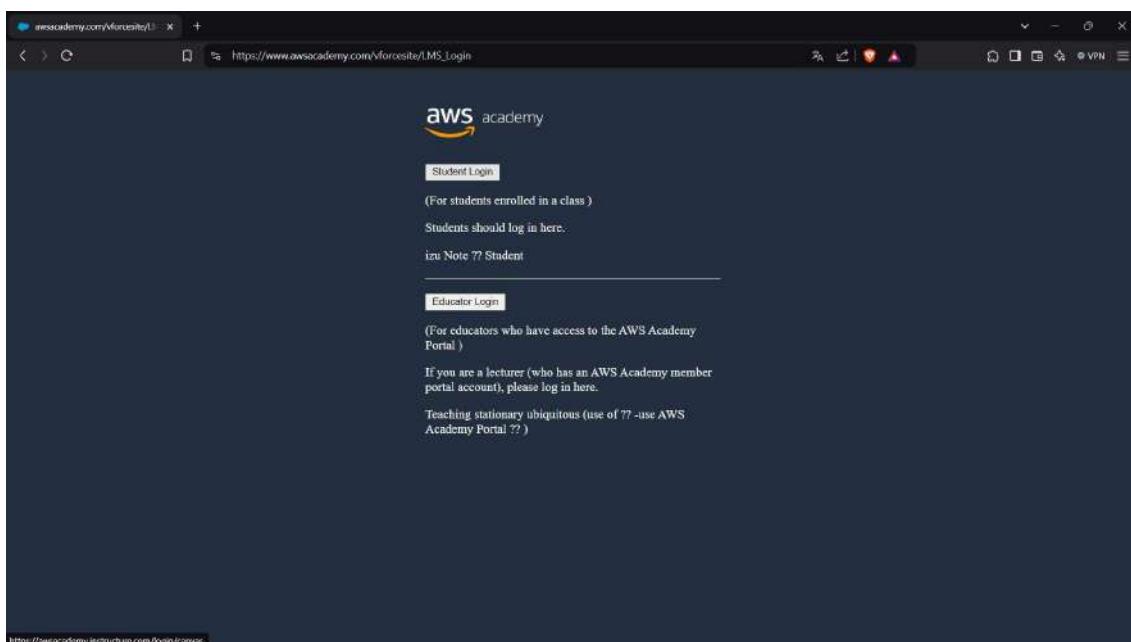
Make sure the dot is turned to red.

8 C. MAVEN WEB PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE

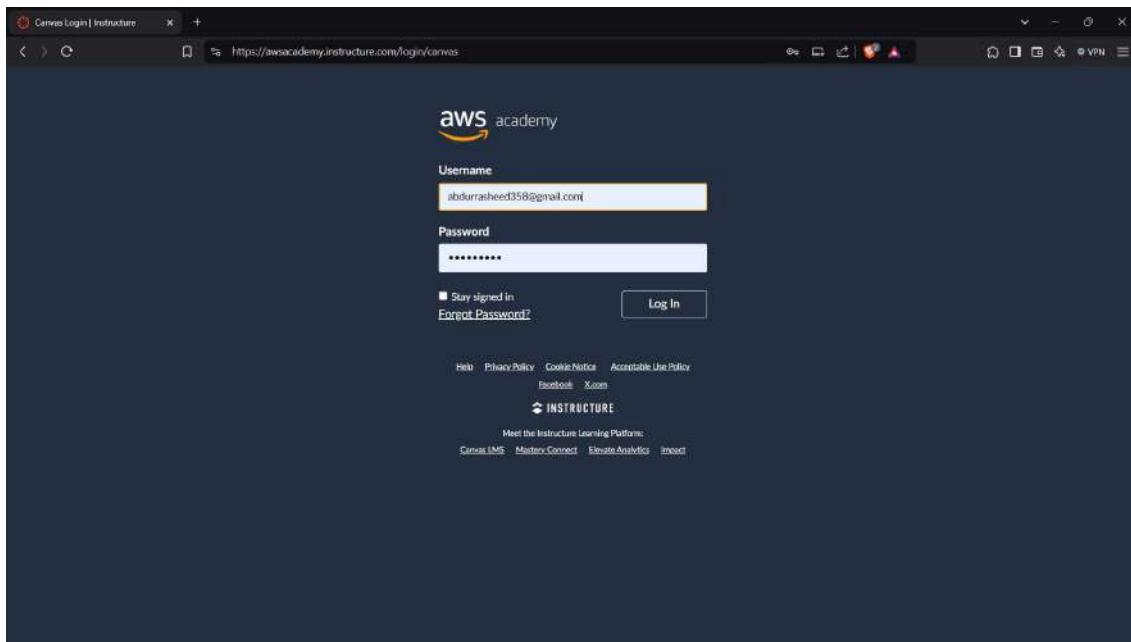
Search for aws Academy Login.



Check for Untitled.



Click on Student Login.



Enter your details and login.

A screenshot of the AWS Academy Dashboard at https://awsacademy.instructure.com. The dashboard has a dark theme. On the left, a sidebar menu includes 'Account', 'Dashboard', 'Courses', 'Calendar', 'Inbox', 'History', and 'Help'. The main area shows a purple card for 'AWS Academy Learner Lab [90659] ALLy2EN-US-LT113-90659'. To the right, there are sections for 'To Do' (Nothing for now), 'Recent Feedback' (Nothing for now), and a 'View Grades' button. At the bottom, there are links for 'Privacy Policy', 'Cookie Notice', 'Acceptable Use Policy', 'Facebook', and 'X.com'. A 'INSTRUCTURE' link is also present at the bottom left.

Click on AWS Academy Learner Lab->Modules.

The screenshot shows a web browser window for the AWS Academy Learner Lab. The URL is https://awsacademy.instructure.com/courses/90659/modules. The page displays the 'AWS Academy Learner Lab' module. On the left, there is a sidebar with icons for Home, Modules, Discussions, Grades, Lucid (Whiteboard), Courses, Calendar, Inbox, History, and Help. The 'Modules' icon is highlighted. The main content area shows a 'Learn how to effectively use the AWS Academy Learner Lab' section with a 'Module Knowledge Check' card (100 pts, Score at least 70.0). Below this is a 'Launch AWS Academy Learner Lab' section with a link to 'Launch AWS Academy Learner Lab'. Further down, there is a 'AWS Academy Learner Lab Resources' section containing links to 'Demo - How to Access Learner Lab', 'Demo - General Troubleshooting Tips', 'Demo - How to Launch Services through AWS Console', and 'Learner Lab Activity - Amazon Q Developer'. At the bottom, there is a 'Feedback Survey' section.

Click on Launch AWS Academy Learner Lab.

The screenshot shows a web browser window titled 'Launch AWS Academy Learner Lab'. The URL is https://awsacademy.instructure.com/courses/90659/modules/items/8311185. The page displays the 'Launch AWS Academy Learner Lab' item. The sidebar on the left is identical to the previous screenshot. The main content area features a large blue downward-pointing arrow with an orange swoosh underneath it, indicating the next step. At the bottom, there are navigation buttons for 'Previous' and 'Next'.

Click on Start Lab.

Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 instances
SSH Access to EC2 instances
SSH Access from Windows
SSH Access from a Mac

Instructions last updated: 2024-12-18

After the red dot turns to green, click on it.

Recently visited: EC2

Applications (0): Create application

Welcome to AWS: Getting started with AWS. Learn the fundamentals and find valuable information to

AWS Health: Open issues 0 Past 7 days

Cost and usage: Current month costs \$0.02 Cost (\$)

Click on EC2.

The screenshot shows the AWS Lambda function configuration page for the function 'Launch AWS Academy Learner Lab'. The left sidebar lists various AWS services: Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IPs. The main content area displays the Lambda function details, including the function name, runtime (Node.js 14.x), memory (128 MB), timeout (3 minutes), and triggers (CloudWatch Events). It also shows the code size (1.16 MB) and the last deployment (1 minute ago). The 'Handler' field contains the code snippet provided in the image. The 'Configuration' tab is selected, showing the 'Environment' section with the environment variable 'LAMBDA_TASK_ROOT' set to '/var/task'. The 'Logs' tab is also visible.

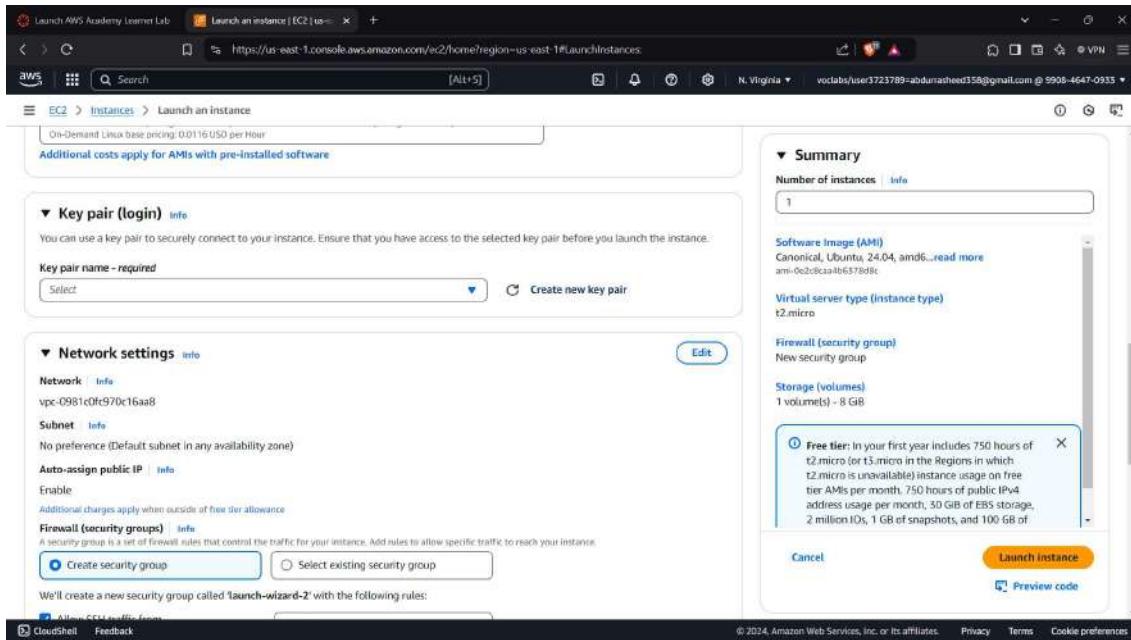
Click Launch Instance.

The screenshot shows the AWS Launch Instances page. In the 'Name and tags' section, the name 'MavenWebProjectServer' is entered. In the 'Application and OS Images (Amazon Machine Image)' section, the search bar contains 'Search our full catalog including 1000s of application and OS images'. Below the search bar, there are sections for 'Recent' and 'Quick Start' AMIs, including options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A sidebar on the right provides a summary of the instance configuration, including the number of instances (1), software image (Amazon Linux 2023 AMI 2023.6.2), virtual server type (t2.micro), firewall (New security group), and storage (1 volume(s) - 8 GB). The 'Launch instance' button is highlighted at the bottom right.

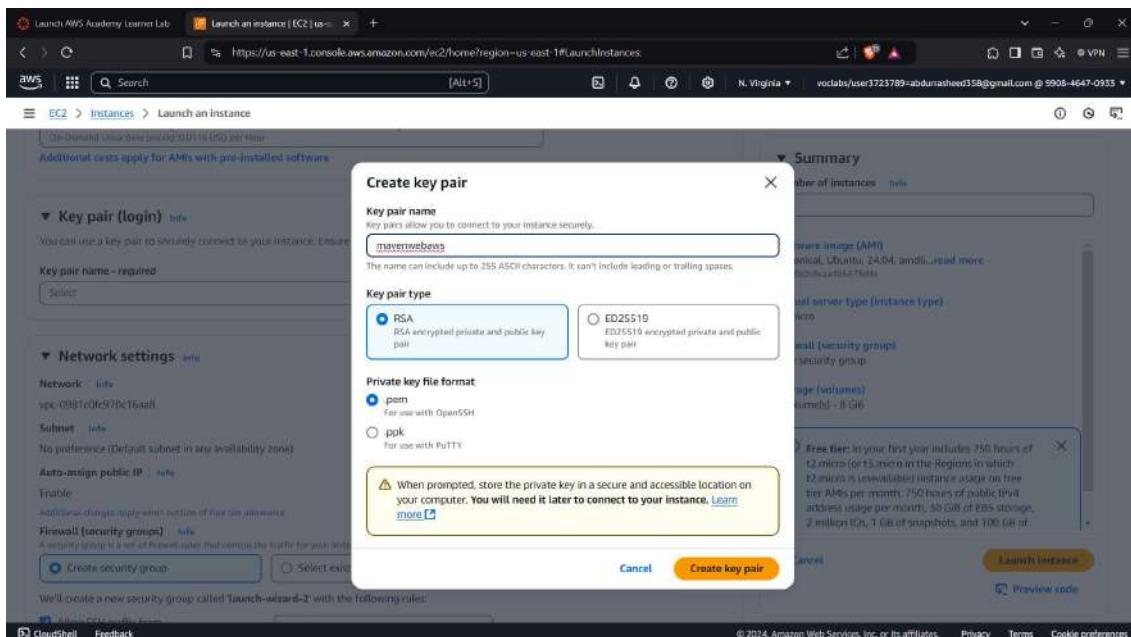
Name: Enter a descriptive name, e.g., MavenWebProjectServer.

AMI: Select Ubuntu Server (Free Tier Eligible).

Instance Type: Choose t2.micro.



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



Give name to Key Pair.

Click on Create Key Pair and save it in a folder.

Network settings

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64... [read more](#)

Virtual server type (instance type): t2.micro

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

Configure storage

Network Settings: Enable Allow HTTP/HTTPS traffic.

Configure storage

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64... [read more](#)

Virtual server type (instance type): t2.micro

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

Advanced details

Storage: Use the default size (8 GB).

Click [Launch Instance](#).

The screenshot shows the AWS Lambda console with a green success message: "Successfully initiated launch of instance i-0a37769fbfd0763c". Below the message, there's a "Launch log" link. A "Next Steps" section contains several options: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", "Create EBS snapshot policy", "Manage detailed monitoring", "Create Load Balancer", "Create AWS budget", and "Manage CloudWatch alarms". At the bottom, there are links for "CloudShell", "Feedback", and copyright information: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Instance is initiated.

The screenshot shows the AWS EC2 Instances page. On the left, a navigation menu includes "Instances", "Images", "Elastic Block Store", and "Network & Security". The main area displays a table titled "Instances (1/1) info" with one row: "MavenWebPro... i-0a37769fbfd0763c Running t2.micro Initializing". Below the table, a detailed view for instance "i-0a37769fbfd0763c (MavenWebProjectServer)" is shown. The "Details" tab is selected, displaying information such as "Public IPv4 address: 3.85.30.94", "Instance state: Running", and "Private IP DNS name (IPv4 only)". The status bar at the bottom indicates "Wait for the status to change to 'Running.'"

Wait for the status to change to "Running."

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IPs. The main area displays 'Instances (1/1) Info' for instance i-0a37769fbfd0763c, which is named 'MavenWebProjectServer'. The instance details include: Instance ID (i-0a37769fbfd0763c), Public IPv4 address (3.85.30.94), Instance state (Running), Private IP DNS name (ip-172-31-26-70.ec2.internal), and Instance type (t2.micro). There are also sections for Public IPv4 DNS (ec2-3-85-30-94.compute-1.amazonaws.com) and Elastic IP addresses.

Note down the Public IP Address from the EC2 dashboard. 3.85.30.94

The screenshot shows the 'Connect to instance' dialog for instance i-0a37769fbfd0763c. The dialog has tabs for EC2 Instance Connect, Session Manager, SSH client (selected), and EC2 serial console. Under the SSH client tab, it says 'Connect to your instance i-0a37769fbfd0763c (MavenWebProjectServer) using any of these options'. It provides instructions: 1. Open an SSH client, 2. Locate your private key file, 3. Run this command if necessary, and 4. Connect to your instance using its Public DNS. A command line shows 'ssh -i "mavenwebaws.pem" ubuntu@ec2-3-85-30-94.compute-1.amazonaws.com'. A note at the bottom says '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.' A 'Cancel' button is at the bottom right.

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

ssh -i "mavenwebaws.pem" ubuntu@ec2-3-85-30-94.compute-1.amazonaws.com

The screenshot shows a Windows File Explorer window titled 'AWSExample3'. The path is 'Desktop > AWSExample3'. Inside 'AWSExample3', there is a file named 'mavenwebaws.pem'. The file is a PEM File, 2 KB in size, and was modified on 27-12-2024 20:15. The file icon is a small square with a white 'm' on it.

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

Copy the path of the folder.

```
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\abdur\OneDrive\Desktop\AWSExample3
PS C:\Users\abdur\OneDrive\Desktop\AWSExample3>
```

Open PowerShell (Windows) or Terminal (Mac/Linux) on your computer->Run As Administrator.

Navigate to the folder where your .pem file is saved using the cd command.

```
ubuntu@ip-172-31-26-70:~$ PS C:\Users\abdur\OneDrive\Desktop\AWSExample3> ssh -i "mavenwebaws.pem" ubuntu@ec2-3-85-30-94.compute-1.amazonaws.com
The authenticity of host 'ec2-3-85-30-94.compute-1.amazonaws.com (3.85.30.94)' can't be established.
ED25519 key fingerprint is SHA256:YcTdfVcaq7nll+D+/ByVOXO]NxrlifJwLLjYLQQigc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-85-30-94.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.0.0-1018-aws x86_64)

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

System information as of Fri Dec 27 14:57:19 UTC 2024

System load: 0.0          Processes:           184
Usage of /: 24.7% of 6.71GB  Users logged in:    0
Memory usage: 20%          IPv4 address for enX0: 172.31.26.70
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-26-70:~$
```

Paste the SSH command and press Enter.

If prompted, type "yes" to confirm the connection.

```

ubuntu@ip-172-31-26-70:~$ sudo apt update
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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
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 [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 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 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [761 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/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 [572 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [118 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 [948 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [552 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Component [208 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.7 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 [1044 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 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 Translation-en [111 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7208 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]

```

Update the system:

sudo apt update

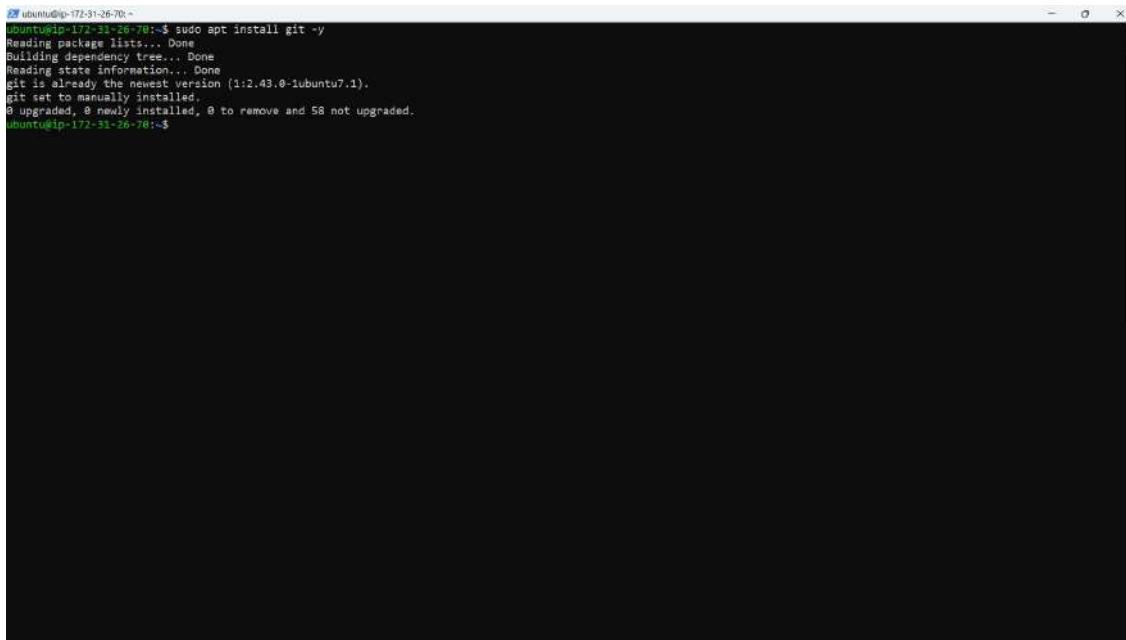
```

ubuntu@ip-172-31-26-70:~$ 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 containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown autofs-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 containerd 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 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 runic amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 containerd amd64 1.7.19+really1.7.12-ubuntu4.2 [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 1s (96.9 MB/s)
Preconfiguring packages...
Selecting previously unselected package pigz.
(Reading database ... 76801 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-ubuntu4.2_amd64.deb ...
Unpacking containerd (1.7.19+really1.7.12-ubuntu4.2) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../4-dns-root-data_2023112702-willsync1_all.deb ...
Unpacking dns-root-data (2023112702-willsync1) ...
Selecting previously unselected package dnsmasq-base.
Preparing to unpack .../5-dnsmasq-base_2.90-2build2_amd64.deb ...
Unpacking dnsmasq-base (2.90-2build2) ...

```

Install Docker:

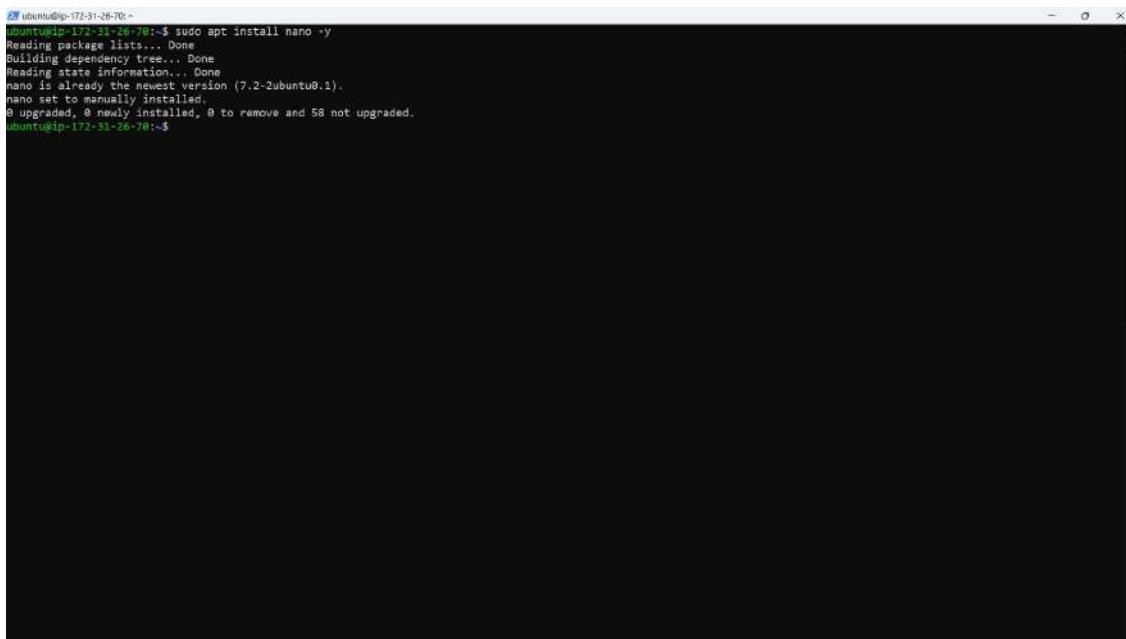
sudo apt-get install docker.io -y



```
ubuntu@ip-172-31-26-78:~$ 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-172-31-26-78:~$
```

Install Git:

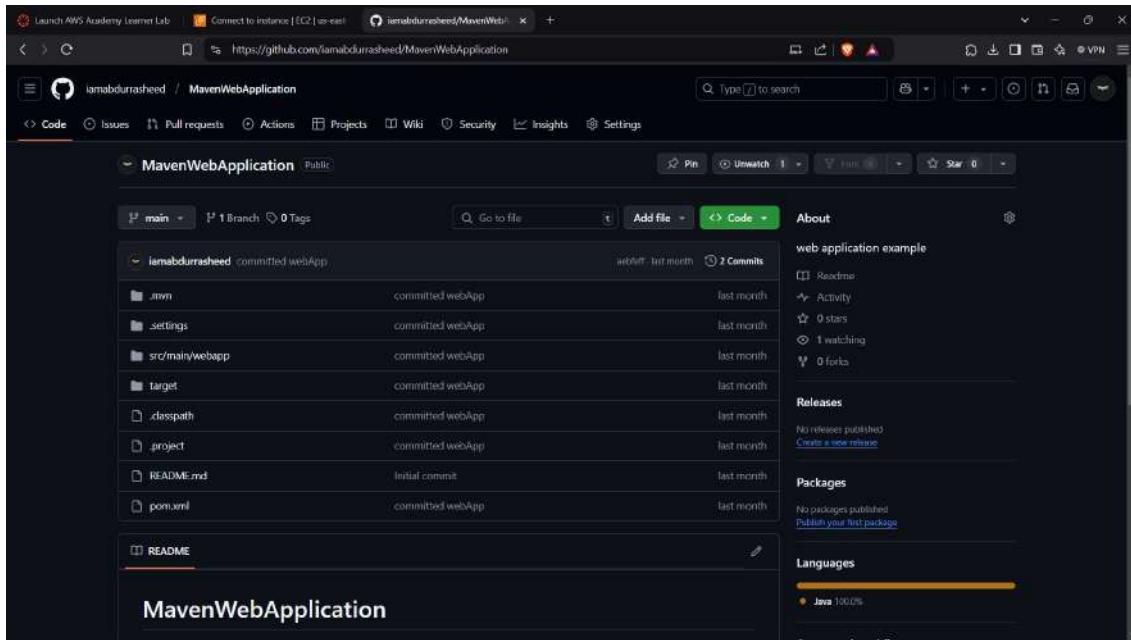
```
sudo apt install git -y
```



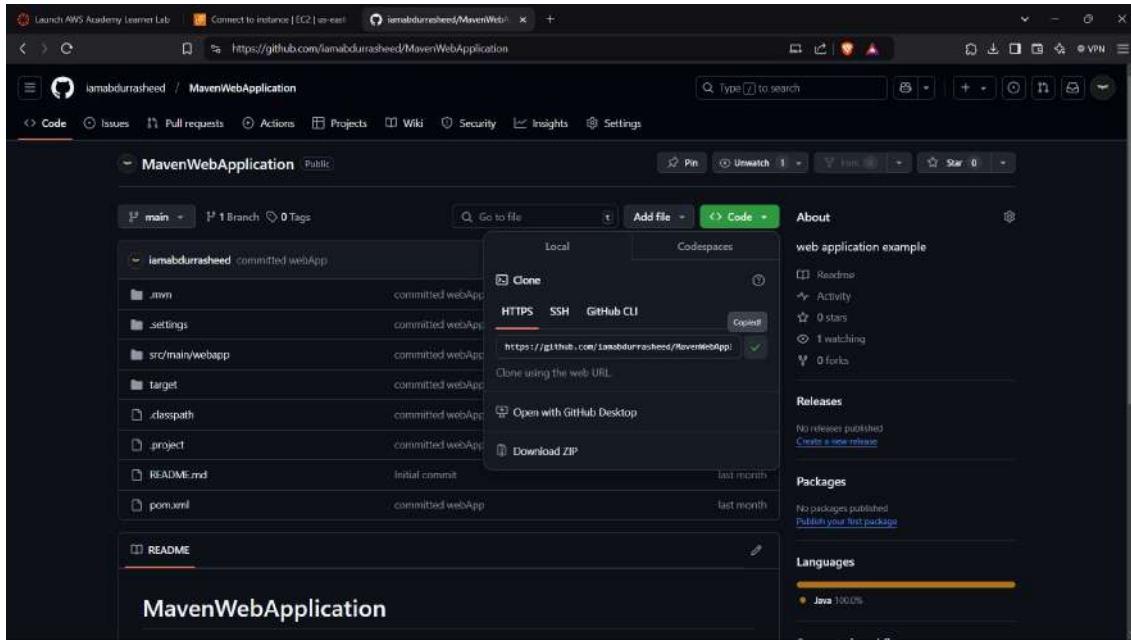
```
ubuntu@ip-172-31-26-78:~$ sudo apt install nano -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-2ubuntu0.1).
nano set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-26-78:~$
```

Install Nano (text editor):

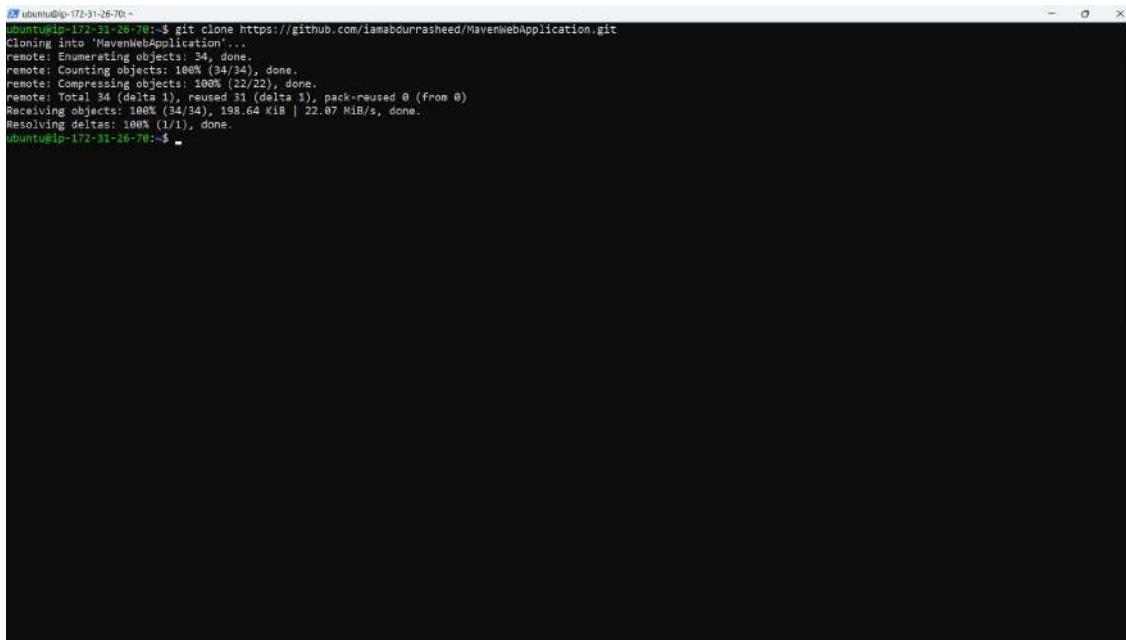
```
sudo apt install nano -y
```



Go to your GitHub repository.



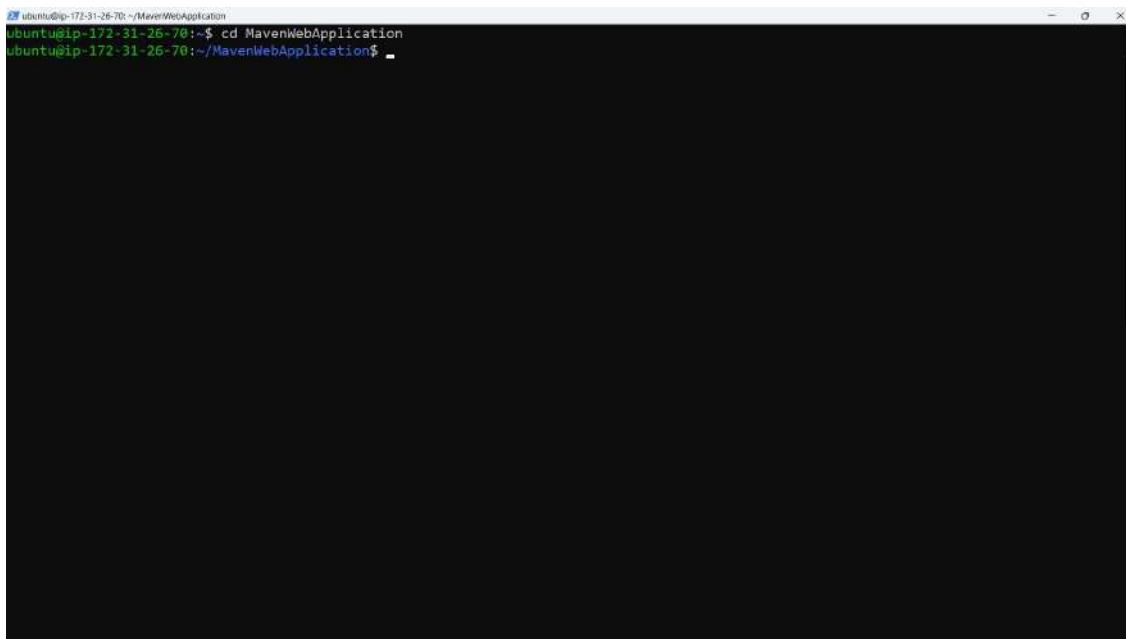
Click Code > HTTPS, and copy the URL.



```
ubuntu@ip-172-31-26-70:~$ git clone https://github.com/iamsbourrasheed/MavenWebApplication.git
Cloning into 'MavenWebApplication'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 34 (delta 1), reused 31 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 198.64 KiB | 22.07 MiB/s, done.
Resolving deltas: 100% (1/1), done.
ubuntu@ip-172-31-26-70:~$
```

Clone the repository:

git clone “repo URL”.



```
ubuntu@ip-172-31-26-70:~/MavenWebApplication$ cd MavenWebApplication
ubuntu@ip-172-31-26-70:~/MavenWebApplication$
```

Navigate to the project folder:

cd “Project Folder”.

```
ubuntu@ip-172-31-26-70:~/MavenWebApplication
GNU nano 7.2
FROM tomcat:9-jdk21
COPY target/*.war /usr/local/tomcat/webapps/
[ Wrote 3 lines.]
```

File menu: Open, Save, Exit
Edit menu: Undo, Redo, Cut, Copy, Paste, Select All, Find, Replace
View menu: Location, Go To Line, Set Mark
Help menu: About, Exit

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

For JDK 21:

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

Save and exit Nano: Press Ctrl + O, then Enter, and Ctrl + X.

```
ubuntu@ip-172-31-26-70:~/MavenWebApplication$ sudo docker build -t mavenwebproject .
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 506.9kB
Step 1/2 : FROM tomcat:9-jdk21
9-jdk21: Pulling from library/tomcat
de44ab265507a: Pull complete
4c2afdf91a87d: Pull complete
89e9bbcfaf697: Pull complete
11be3e613582: Pull complete
1b9d1e181a2a: Pull complete
fc68c0117916: Pull complete
4f4fb700ef54: Pull complete
c9f74ce6014e: Pull complete
Digest: sha256:d9f92b415416c22e8ada5e7179bfff7b0cded9687732068906292a6f960b3f917
Status: Downloaded newer image for tomcat:9-jdk21
--> 39642322f89a
Step 2/2 : COPY target/*.war /usr/local/tomcat/webapps/
--> cabab751ffcd9
Successfully built cabab751ffcd9
Successfully tagged mavenwebproject:latest
ubuntu@ip-172-31-26-70:~/MavenWebApplication$
```

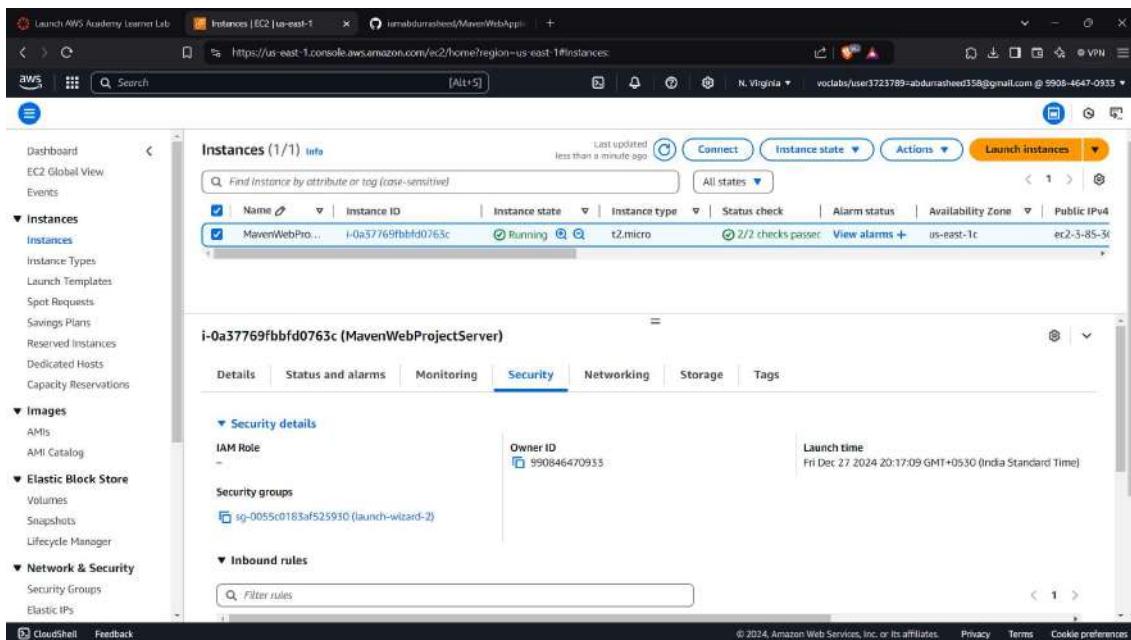
Build the Docker image:

```
sudo docker build -t maven-web-project .
```

```
ubuntu@ip-172-31-26-70:~/MavenWebApplication$ sudo docker run -d -p 9090:8080 mavenwebproject
8a4c27a66f483bf0d291485eae5e0c555605aae9e9267461bed235f107b55d3e
ubuntu@ip-172-31-26-70:~/MavenWebApplication$
```

Run the container:

```
sudo docker run -d -p 9090:8080 maven-web-project
```



In the AWS EC2 dashboard, go to Security and click the Security Group ID.

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#function:arn:aws:lambda:us-east-1:990846470933:function:lambdafunctionname>. The page displays the function configuration, including the Handler, Runtime, and Configuration tabs. The Handler is set to `lambdafunctionname::lambda_handler`, the Runtime is Node.js 18.x, and the Configuration tab shows the environment variables and triggers.

Click on Edit Inbound Rules.

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#function:arn:aws:lambda:us-east-1:990846470933:function:lambdafunctionname>. The page displays the function configuration, including the Handler, Runtime, and Configuration tabs. The Handler is set to `lambdafunctionname::lambda_handler`, the Runtime is Node.js 18.x, and the Configuration tab shows the environment variables and triggers.

Add an inbound rule.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-077e7fdf7ceba6eac	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-01f42c04c0b517f5a	SSH	TCP	22	Custom	0.0.0.0/0
sgr-0722eae24c4c5874	HTTPS	TCP	443	Custom	0.0.0.0/0
-	Custom TCP	TCP	9090	Anyw...	0.0.0.0/0

Add rule

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

Save rules

Type: Custom TCP

Port Range: 9090

Source: Anywhere (0.0.0.0/0) or your IP.

Save the changes.

Inbound security group rules successfully modified on security group (sg-0055c0183af525930 | launch-wizard-2)

sg-0055c0183af525930 - launch-wizard-2

Details

Security group name	Security group ID	Description
launch-wizard-2	sg-0055c0183af525930	launch-wizard-2 created 2024-12-27T14:33:29.592Z
Owner	Inbound rules count	Outbound rules count
990846470933	4 Permission entries	1 Permission entry

Inbound rules

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0981709a265ba96e8	IPv4	Custom TCP	TCP	9090
-	sgr-077e7fdf7ceba6eac	IPv4	HTTP	TCP	80
-	sgr-01f42c04c0b517f5a	IPv4	SSH	TCP	22

The changes are successfully saved.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar navigation includes: Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IPs. The main content area displays a table titled 'Instances (1/1)'. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. One row is shown for 'MavenWebPro...', which is 'Running' on an 't2.micro' instance in 'us-east-1c' with a public IP of 'ec2-5-85-34'. Below the table, a detailed view for instance 'i-0a37769fbfd0763c (MavenWebProjectServer)' is expanded. It shows details like Instance ID (i-0a37769fbfd0763c), Instance state (Running), and Private IP DNS name (ip-172-51-26-70.ec2.internal). A tooltip over the Public IPv4 address '3.85.30.94' indicates it can be copied.

Copy the Public IP Address.

The screenshot shows a browser window with the URL 'http://3.85.30.94:9090/Maven/WebApp/'. The page content is 'Hello World!'

Open a browser and navigate to:

<http://Public-IP:9090/ProjectName>

```
ubuntu@ip-172-31-26-70:~/MavenWebApplication$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
8a4c27a66f48      mavenwebproject   "catalina.sh run"   41 minutes ago   Up 40 minutes   0.0.0.0:9090->8080/tcp, :::9090->8080/tcp   mavenweb
al_knuth
ubuntu@ip-172-31-26-70:~/MavenWebApplication$ sudo docker stop 8a4c27a66f48
8a4c27a66f48
ubuntu@ip-172-31-26-70:~/MavenWebApplication$
```

sudo docker ps to display running containers.

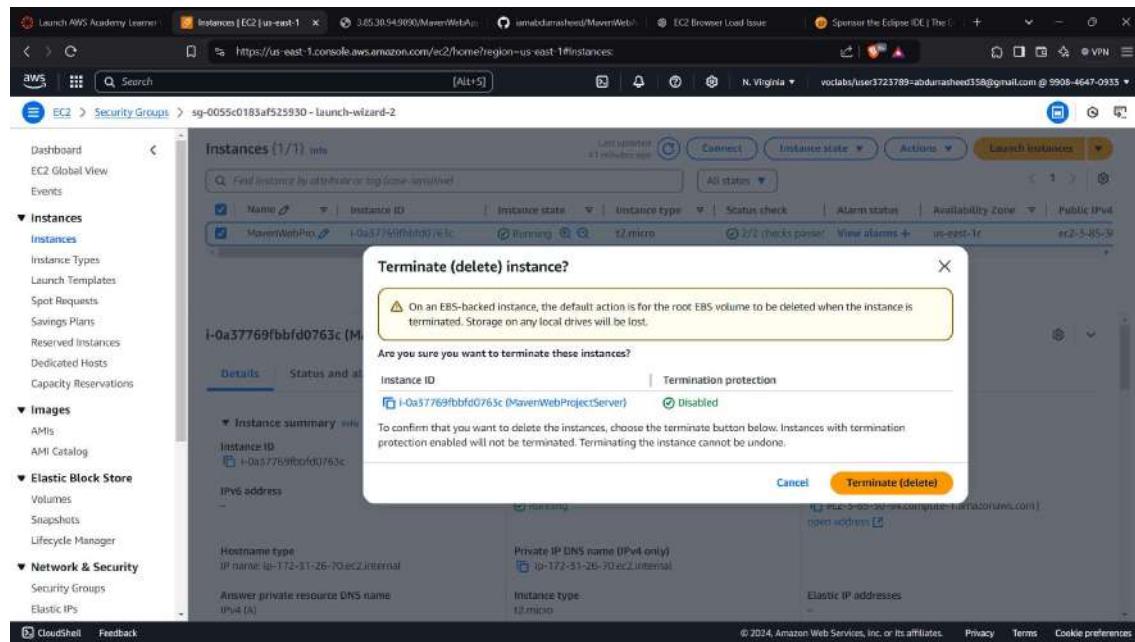
Copy the container ID.

Stop the Docker container:

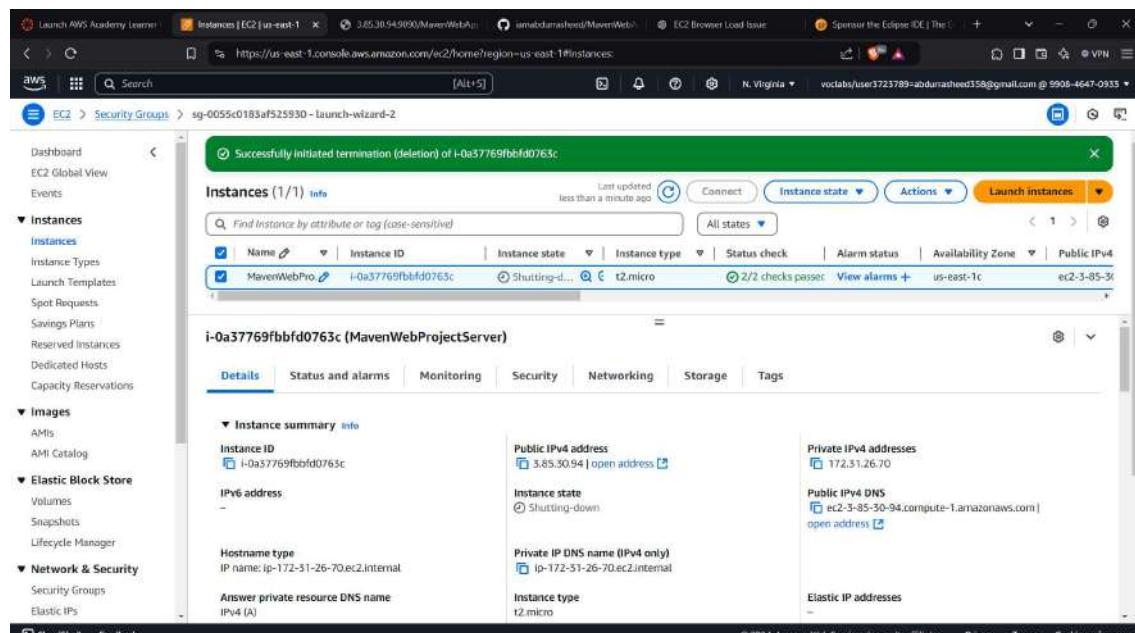
sudo docker stop <Container ID>

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IPs. The main area displays a table titled 'Instances (1/1)'. It shows one instance named 'MavenWebPro' with the ID 'i-0a37769fbfffd0763c'. The instance is listed as 'Running' in the 'Instance state' column. To the right of the table, there's a 'Actions' dropdown menu with several options: Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate (delete) instance. The 'Terminate (delete) instance' option is currently selected. Below the table, there's a detailed view for the instance 'i-0a37769fbfffd0763c (MavenWebProjectServer)'. This view includes sections for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there's an 'Instance summary' section with fields like Instance ID, Public IPv4 address, Private IPv4 addresses, and so on.

In the EC2 dashboard, go to Instance State and select Terminate Instance.



Click on Terminate(delete)



The instance is terminated successfully.

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has tabs for Home, Modules, Discussions, Grades, and Lucid (Whiteboard). A central terminal window displays a command-line session: `www_n_3942766@unweb1560836:~$`. To the right is a sidebar titled "Learner Lab" with sections for Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, Preserving your budget, Accessing EC2 instances, SSH Access to EC2 instances, SSH Access from Windows, and SSH Access from a Mac. At the bottom right of the sidebar is the text "Instructions last updated: 2024-12-18". Navigation buttons at the bottom include "Previous" and "Next".

End Lab to avoid charges.

The screenshot shows the same AWS Academy Learner Lab interface as above, but with a modal dialog box in the center asking "Are you sure you want to end the lab?". Below the dialog are two buttons: "Yes" and "No". The background content is identical to the first screenshot, including the terminal session and the "Learner Lab" sidebar with its various links and update information.

Click on Yes.

b

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with navigation links: Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has a top navigation bar with tabs like Launch AWS Academy Learner Lab, Instances [EC2] u-end-1, 3.65.30.94.900/MavenWeb..., imediarashed/MavenWeb..., EC2 Browser Load Issue, Sponsor the Eclipse IDE The..., and a VPN icon. Below the bar is a breadcrumb trail: ALLv2EN-US... > Modules > AWS Academy Learner Lab > Launch AWS Academy Learner Lab. The central part of the screen features a terminal window titled 'AWS' with a red dot, showing the command line: '***_K_3942766@runweb156036:~\$'. To the right of the terminal is a 'Learner Lab' section with a dropdown menu set to 'EN-US'. This section contains a list of links: Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, Preserving your budget, Accessing EC2 instances, SSH Access to EC2 instances, SSH Access from Windows, and SSH Access from a Mac. At the bottom of the 'Learner Lab' section is a note: 'Instructions last updated: 2024-12-18'. Navigation buttons at the bottom include '< Previous' and 'Next >'.

The green dot turned to red indicates lab is ended.