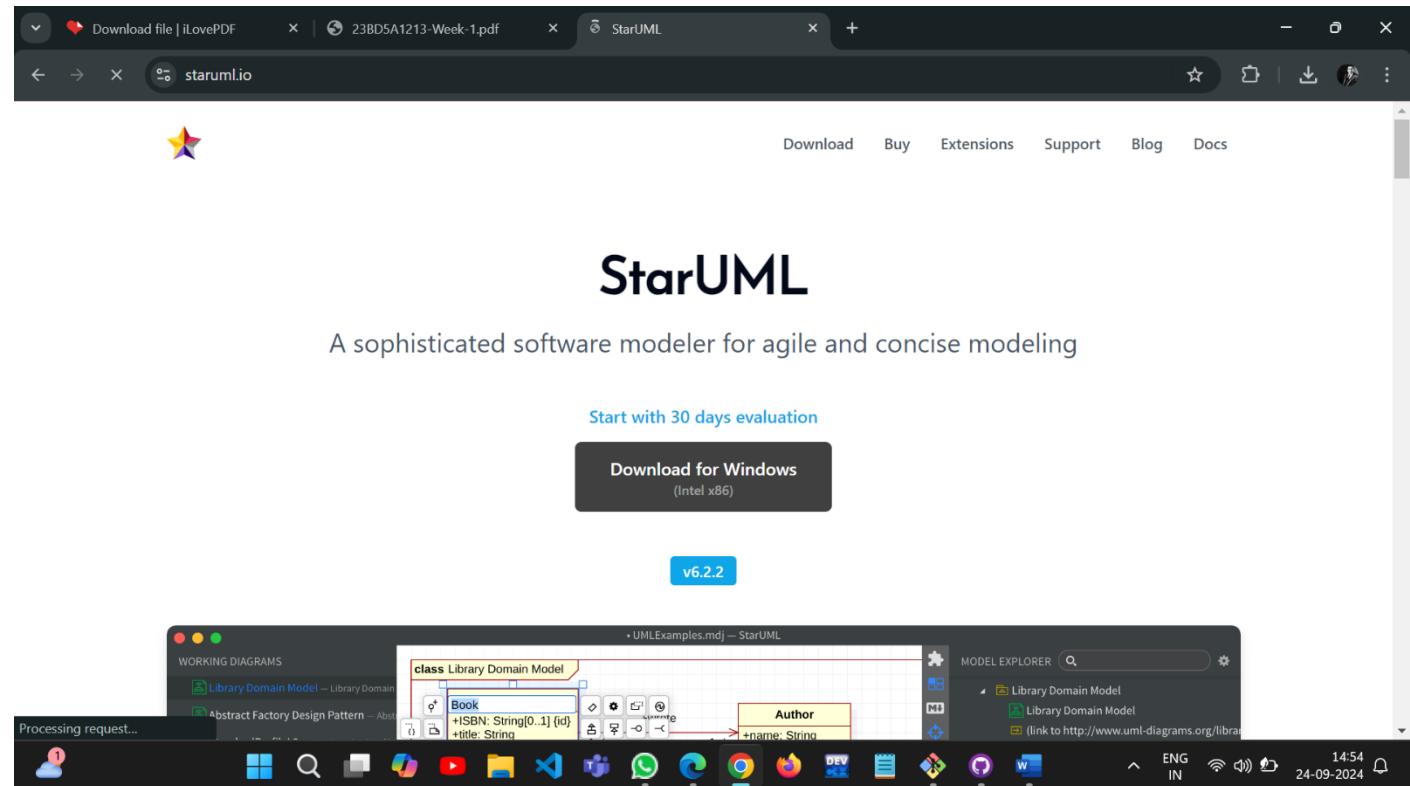


Experiment - 1

INSTALLATION OF STAR UML

STEP 1:

Download and install star UML from <https://staruml.io/>



STEP 2:

Click on download for windows button based on the your system architecture

STEP 3:

Install the download star UML software

STEP 4:

Explore with tool bar options



INSTALLATION OF GIT-SCM

STEP 1:

Download and install Git from the <http://git-scm.com/>

The screenshot shows the official Git website (<http://git-scm.com/>). At the top, there's a search bar and a link to exit full screen. The main content area features a large graphic of books connected by red and blue lines, symbolizing distributed version control. Below this, there are four main navigation links: "About" (with a gear icon), "Documentation" (with a book icon), "Downloads" (with a download arrow icon), and "Community" (with a speech bubble icon). A sidebar on the left contains links for "Pro Git" (with a book icon) and "GUI Clients" (with a monitor icon). On the right, a large monitor displays a teal box with the text "Latest source Release 2.46.2" and a "Download for Windows" button. Below the monitor are links for "Windows GUIs" and "Tarballs".

STEP 2:

Select the Git software for the required OS

This screenshot shows the "Downloads" section of the Git website for Windows. It features a sidebar with links for "About", "Documentation", "Downloads" (selected), "GUI Clients", "Logos", and "Community". The main content area is titled "Download for Windows" and includes a link to download the latest version (2.46.0). It lists several download options: "Standalone Installer", "32-bit Git for Windows Setup", "64-bit Git for Windows Setup", "Portable ("thumbdrive edition")", "32-bit Git for Windows Portable", and "64-bit Git for Windows Portable". Below these, there's a section for "Using winget tool" with a command-line example and a note about the current source code release being version 2.46.0. A "Now What?" section at the bottom provides instructions for starting to use Git.

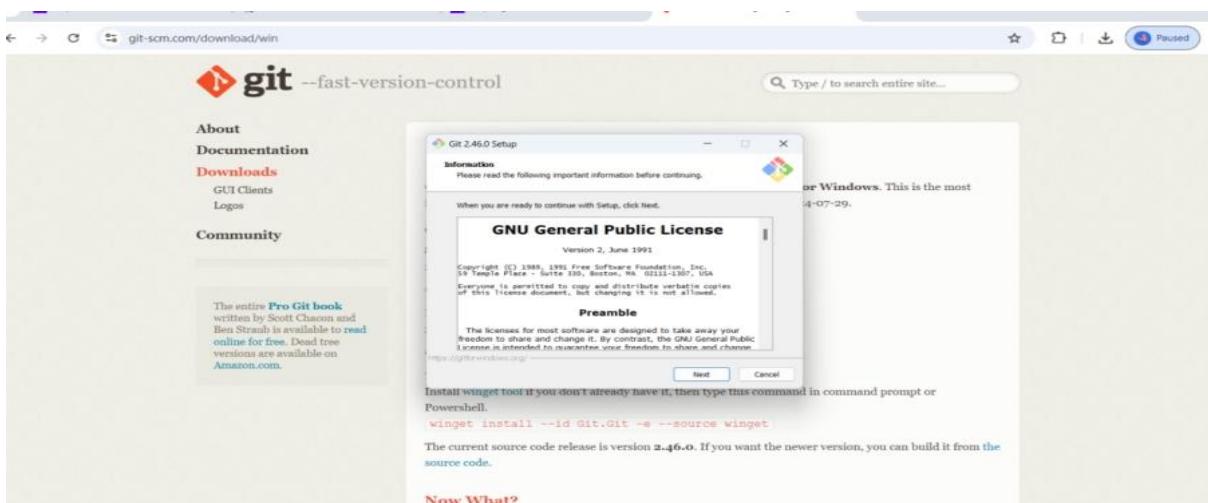
STEP 3:

Install the download Git software



STEP 4:

While installing select the option use Git from Git Bash only option and click Next



STEP 5:

You will now find the Git Bash after successful installation

```
MINGW64:/c/Users/singi
singi@BUNNY MINGW64 ~
$
```

The screenshot shows a Windows desktop environment. At the top is a terminal window titled 'MINGW64:/c/Users/singi' with the command 'singi@BUNNY MINGW64 ~' and a single '\$'. Below it is a dark-themed taskbar with icons for File Explorer, Task View, and many other pinned and running applications. The system tray shows the date '24-09-2024' and time '15:05'.

CREATION OF GIT-HUB ACCOUNT

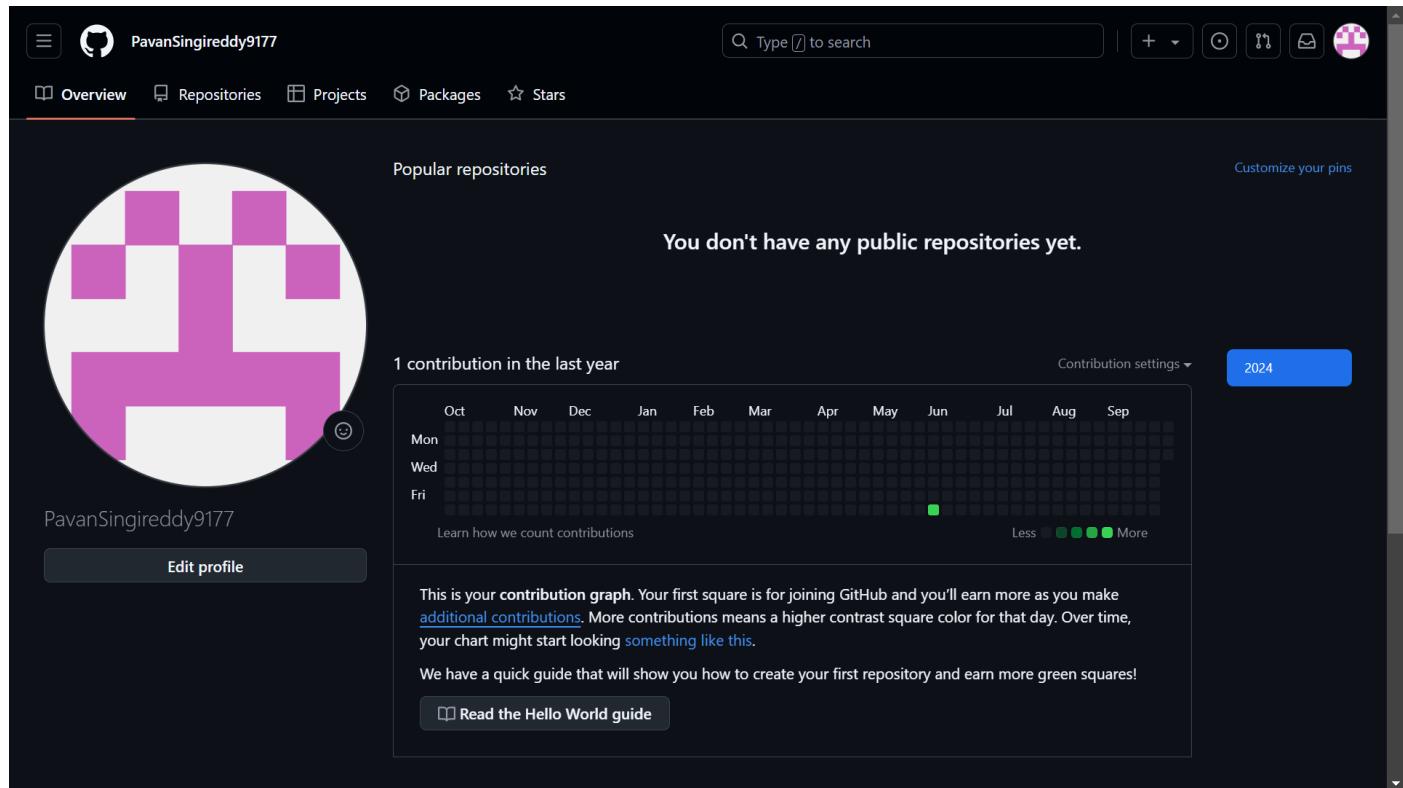
STEP 1:

Browse the website <http://github.com> and click on Signup

The screenshot shows a GitHub profile page for the user 'PavanSingireddy9177'. The profile picture is a pink and white pixelated logo. The page includes a search bar, navigation links for Overview, Repositories, Projects, Packages, and Stars, and a 'Customize your pins' button. A message states 'You don't have any public repositories yet.' Below this is a 'Contribution graph' section showing '1 contribution in the last year' with a timeline from Oct to Sep. The graph has a single green square for a contribution on Friday, October 6th. A note explains how contributions are counted, mentioning 'Less' and 'More' scales. A 'Read the Hello World guide' button is at the bottom.

STEP 2:

Enter your email address with which you want to access git-hub and also create a password for the same ,select Notifications received through email -----'

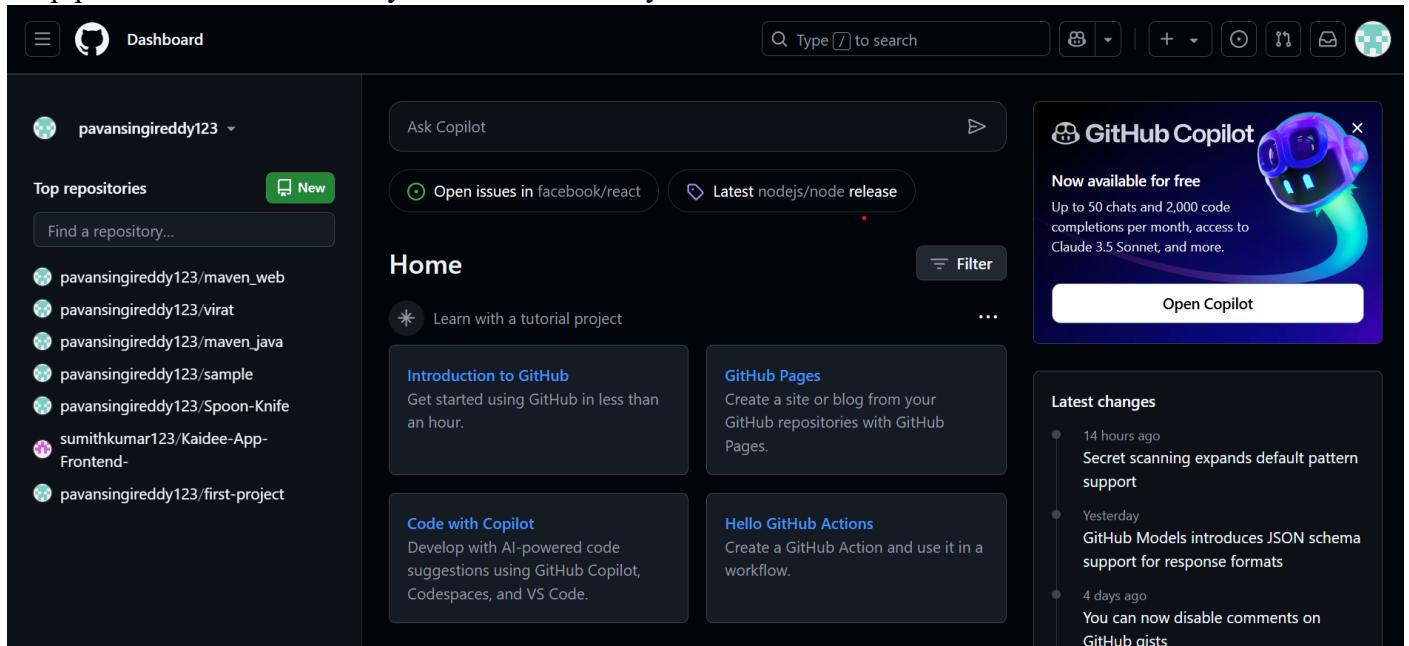


STEP 3:

Complete the verification process by selecting the spiral galaxy in the given set of images

STEP 4:

Skip personalization And your account is ready



Experiment - 2

Software Requirements Specification

for

ONLINE COURSE REGISTRATION

Submitted To: Savitha Ramesh

Table of Contents

Revision History.....
Problem Statement.....
1.Introduction.....
 1.1 Purpose.....
 1.2 Document Conventions.....
 1.3 Definitions, Abbreviations.....
 1.4 Project Scope.....
2.Overview.....
 2.1 Bus Reservation System.....
3.Overall Description.....
 3.1 Product.....
 3.2 Product Functionality.....
4.Specific Requirements.....
 4.1 System Requirements.....
 4.2 Functional Requirements.....
5.Conclusion.....

Revision History

Name	Date	Reason For Changes	Version
Week-1	15-09-20	SRS Creation (Introduction)	1.0

Problem Statement

The Online Course Reservation System is designed to simplify course registration for students and administrators by providing a user-friendly, web-based platform. Students can easily browse available courses, check prerequisites, and reserve seats with real-time seat availability updates.

Administrators will have tools to manage course offerings, faculty assignments, and enrollment limits, significantly reducing manual errors and improving efficiency. The system will also integrate with existing student information systems, automate waitlisting, and send confirmation emails for successful reservations.

Additionally, the system will enhance accessibility with mobile compatibility, enabling students to register from anywhere. Features like personalized course recommendations, search filters, and calendar synchronization will further streamline the process, allowing students to efficiently plan and manage their academic schedules. This will ultimately reduce administrative overhead and provide a smoother experience for all users involved.

1. INTRODUCTION

1.1 Purpose

This document outlines the software requirements for an online course reservation system. It is intended for designers, developers, and maintainers of the course reservation platform. These requirements were developed in response to the challenges of manual course registration, aiming to improve efficiency, reduce errors, and enhance the user experience.

For students, the system provides a convenient platform to browse and reserve courses, with features like personalized recommendations and mobile access. For administrators, it offers comprehensive tools for managing course offerings, faculty assignments, and enrollment capacities. Ultimately, the system's purpose is to create a seamless, accessible, and efficient registration process, benefiting both students and the academic administration.

1.2 Document Conventions

The Online Course Reservation System is a web-based application that allows students to check course availability, reserve seats, and manage their registrations online. The system is designed for all registered users, providing access to courses and registration features once they have gained authorization from the administrator

1.3 Definitions, Abbreviations

- OCRS – Online Course Reservation System
- CN – Course Number
- Reservation: The process of securing a seat in a specific course.
- Waitlist: A queue where students are placed when a course is full, pending seat availability.
- SID – Student Identification Number
- Report – Summary of a particular course or registration activity
- Admin Staff – Administrative personnel responsible for managing user access, course offerings, and system settings
- Prerequisite: A requirement that must be met before a student can register for a specific course.
- User Access: The permission granted to students, faculty, or administrators to use the system's features based on their role

1.4 Project Scope

The Online Course Reservation System consists of three modules. The first module allows students to inquire about the availability of courses on specific dates, the second module enables them to reserve a seat in a chosen course, and the third module allows for the cancellation of a previously reserved course. The current registration process typically relies on manual methods for enrolling in courses, which can lead to inefficiencies and errors in managing student registrations.

2. Overview

2.1 Online Course Reservation System

During the offline era, course registration posed various challenges for students and educational institutions. Offline methods limited students' ability to explore different options based on their academic criteria and created additional administrative burdens for institutions. Moreover, educational institutions struggled to effectively monitor course enrollment and availability.

Many small and medium educational organizations lack their own online course reservation systems. The Online Course Reservation System is a comprehensive web-based platform that facilitates in-house management of course offerings, registrations, reporting, and other administrative functions associated with student enrollment.

The Online Course Reservation System provides enhanced services to both educational institutions and students, including:

- Capture of student information such as name, address, phone number, and email address
- Course availability and pricing information
- Course rankings or ratings
- Course seating or capacity charts
- Search functionality for easy navigation
- Payment processing for course fees
- Institutional branding, including logos, slogans, and contact information
- A comments and suggestions section for user feedback
- Comprehensive reporting features for enrollment and performance tracking

3. Overall Description

3.1 Product

The Online Course Reservation System is a self-contained platform that manages the activities related to course registrations, including student information and enrollment processes. Various stakeholders are involved in the Online Course Reservation System, including students, faculty, administrators, and support staff, each playing a crucial role in ensuring the smooth operation of the system and enhancing the overall user experience.

3.2 Product Functionality

➤ Registration

When a student seeks to register for a course, the system first checks to see if the student is already registered using their details. If not, a new Student Identification Number (SID) is assigned to the student. The student's information, including date of birth, address, and mobile number, is then captured and stored in the system.

➤ Course Reservation

Once the student logs in with their credentials, they are prompted to select their desired course, including the course code, date, and time. The system checks the database for seat availability in the chosen course. If a seat is available, the reservation is confirmed through e-payment, and the student receives a confirmation with the course details, including seat number, course code, date, and time, along with a digital ticket blueprint.

➤ Course Cancellation

In the event that a student needs to cancel their reservation, the system allows for easy cancellation within a specified time frame. The student can log in to their account and view their current reservations, selecting the course they wish to cancel. Upon cancellation, a confirmation message is sent, and any applicable refunds are processed according to the institution's policies.

➤ Feedback and Support

Students can provide feedback on their course experience through the system, allowing for continuous improvement in course offerings. The system also features a support module where students can submit inquiries or issues they encounter, ensuring prompt assistance from the administration team. This creates a supportive learning environment and enhances student satisfaction.

➤ Report Generation

The system generates various reports, including detailed information regarding the courses offered, enrollment statistics, and student registrations, providing valuable insights for administrators and faculty to manage course offerings effectively.

4. Specific Requirements

4.1 System Requirements

Hardware: Intel Core 2 Duo System or equivalent Minimum 15 TB of storage capa

Software: HTML for structuring web content

CSS for styling and layout design

JavaScript for interactive features and functionality

MySQL for database management and data storage

4.2 Functional Requirements

➤ Registration

- **Assign PIN:** The Online Course Reservation System (OCRS) allows administrators to assign each student a unique Student Identification Number (SID) during the registration process. This ID will be used by the student throughout their academic journey within the system.

➤ Course Reservation

- **Sign In:** Students must first sign in to the system using the username and password they provided during registration. The system will validate the credentials, allowing access only if the information is correct.
- **Check Availability:** Once logged in, students should be able to view all available courses, including relevant details such as course descriptions, timings, and prerequisites.
- **Reserve Course:** If a seat is available, the system will book it under the student's SID. The reservation should be locked to that student until they decide to cancel or until the course begins, preventing double booking.

➤ Course Cancellation

- **Cancel Reservation:** Students should have the ability to cancel their course reservations through their account. Upon cancellation, the seat should become available for other students, and a confirmation of the cancellation should be sent to the student.

➤ Report Generation

- Seat Availability Reports: The OCRS shall generate reports regarding seat availability, including details such as course code, seat number, and occupancy status (occupied/unoccupied).
- Enrollment Statistics: The system will generate reports on overall enrollment data, including the number of students per course and demographic information, helping administrators make informed decisions about course offerings.

➤ Database

- Student-Related Information: Each student shall have the following mandatory information stored: first name, last name, phone number, Student ID, address, postal code, city, username, and password.
- Course-Related Information: Each course should have the following information: course code, course title, number of available seats, course type, and schedule details
- Payment Information: The system must securely store payment-related information for each transaction, ensuring compliance with data protection regulations.

➤ Feedback Mechanism

Course Feedback: Students shall have the ability to provide feedback on courses they have completed. This feedback will be stored in the system and can be used for future improvements to course offerings.

Non-Functional Requirements

➤ Performance Requirements

- **Response Time:** The system shall provide responses within 1 second after processing passenger information to ensure a smooth user experience
- **Capacity:** The system must support simultaneous access for up to 1000 users, accommodating peak usage without degradation in performance.
- **Scalability:** The system should be designed to scale efficiently, allowing for increased user load or additional functionalities without significant rework.

➤ Safety and Security Requirements

- **Student Identification:** The system requires each student to verify their identity using their unique Student Identification Number (SID) during registration and booking processes.

- **Data Modification:** Any modifications (insert, update, delete) in the database shall be synchronized and allowed only by authorized administrators to maintain data integrity.
- **Administrator's Rights:** Administrators shall have comprehensive access to view and modify all information in the system, including user data and course offerings.
- **Data Encryption:** Sensitive information, such as passwords and payment details, must be encrypted during transmission and storage to safeguard against unauthorized access.
- **User Authentication:** The system shall implement robust authentication mechanisms to prevent unauthorized access to student accounts.

➤ Software Quality Attributes

- **Back Up:** The system shall provide regular data backup capabilities to prevent information loss in the event of system failures or outages.
- **Error Logging:** The system shall maintain a log of all errors and exceptions, enabling administrators to monitor, diagnose, and resolve issues effectively.
- **Availability:** The system shall aim for high availability, targeting 99.9% uptime, ensuring users can access the platform at any time without interruption.
- **Usability:** The system shall feature user-friendly interfaces to ensure easy navigation for all users, requiring minimal training or technical support.
- **Maintainability:** The system shall be built with a modular architecture to facilitate easy updates and maintenance, allowing for the addition of new features with minimal downtime.
- **Performance Monitoring:** The system shall include performance monitoring tools to track system responsiveness and load, ensuring proactive management of resources.

5. Conclusion:

In conclusion, the Online Course Reservation System project involves the design and development of a software solution that enables students to register for courses online efficiently. The project's scope includes creating a system that is user-friendly, secure, scalable, and accessible to a diverse range of users, including students, faculty, and administrators. Key functional requirements of the system encompass features such as course registration, seat selection, course cancellation, and secure payment processing. Additionally, non-functional requirements focus on performance, security, accessibility, and overall user experience.

To ensure the success of the project, the development team must also address system constraints, including platform and technology choices, integration with existing educational systems, and ongoing maintenance and support. Collaboration with educational institutions is crucial to ensure that the system aligns with their specific requirements and regulatory standards. By prioritizing user needs and adhering to best practices in software development, the project aims to deliver a robust solution that enhances the course registration experience for all stakeholders involved.

UML DIAGRAMS:

Conceptual Model of UML (Unified Modeling Language)

The conceptual model in UML helps represent the structure and behavior of a system using various diagrams. UML offers a standard way to visualize the design of a system and is widely used in object-oriented software development. It typically involves three main components:

1. **Objects:** These represent real-world entities.
2. **Relationships:** These define how objects interact with each other.
3. **Actions/Behaviors:** This defines the functionality or behavior of the system.

USE CASE DIAGRAM:-

A use case diagram illustrates the interactions between users (actors) and a system. It visually represents how users interact with the system's functionalities.

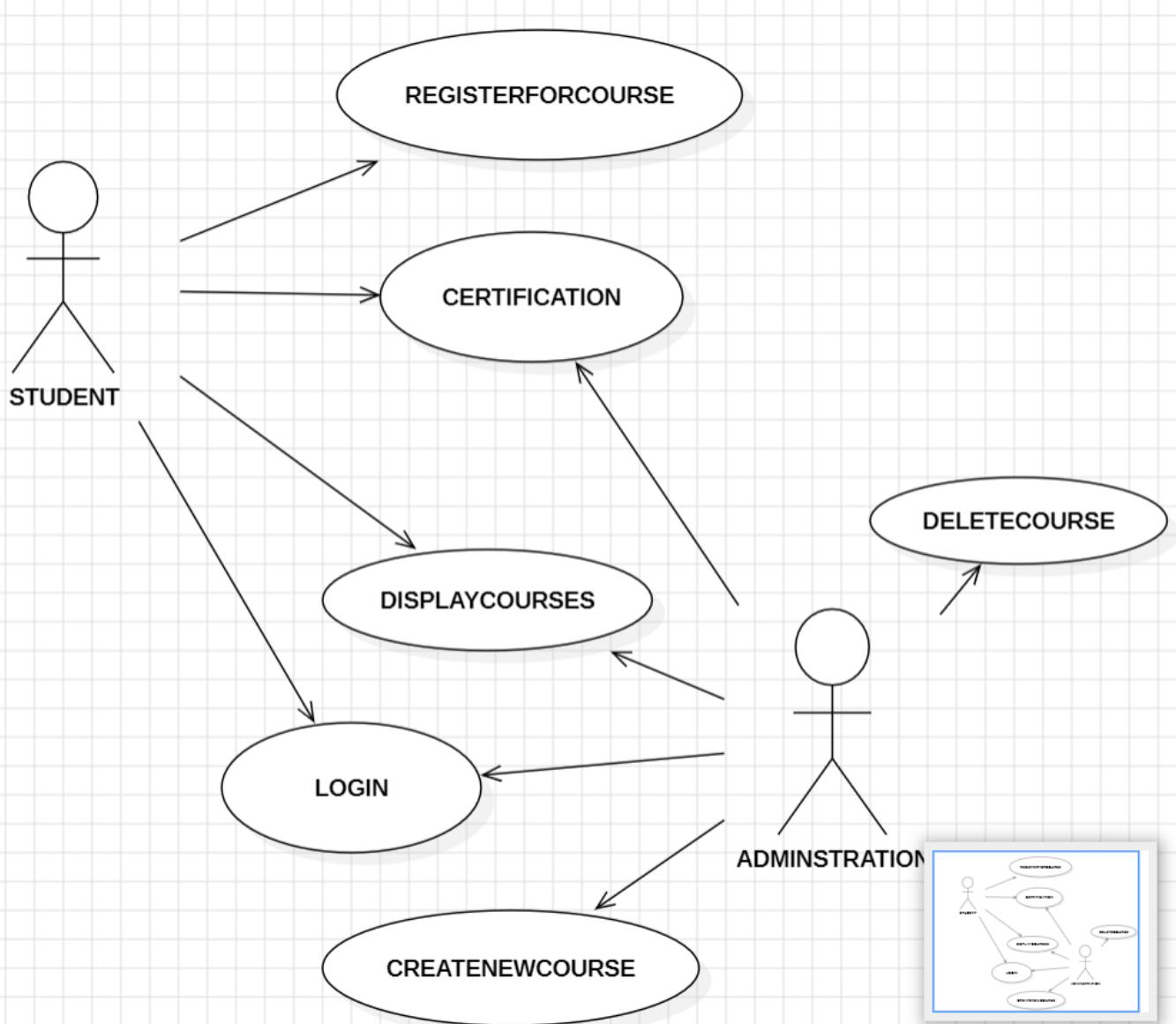
Elements in a Use Case Diagram:

- **Actors:** Represent users or external systems that interact with the system. They can be a person, another system, or an organization.
- **Use Cases:** These represent the specific actions the system can perform. For example, in an "Online Course Reservation System," use cases might include:
 - **Register for a Course**
 - **View Course Details**
 - **Cancel Reservation**
- **Relationships:**
 - **Association:** A line between an actor and a use case that shows interaction.
 - **Include:** A use case that includes the behavior of another use case.
 - **Extend:** A use case that extends the behavior of another use case, typically representing optional actions.
 - **Generalization:** A relationship between actors or use cases where one is a more specific version of the other.

Example (Use Case Diagram for an Online Course Reservation System):

- **Actors:**
 - Student
 - Admin
 - Instructor
- **Use Cases:**
 - Student: Register for Course, View Available Courses, Cancel Reservation.
 - Admin: Add/Remove Courses, View Registrations.
 - Instructor: Update Course Details.

Use Case Diagram:-



CLASS DIAGRAM:-

A class diagram describes the structure of a system by showing its classes, attributes, operations (or methods), and the relationships between objects.

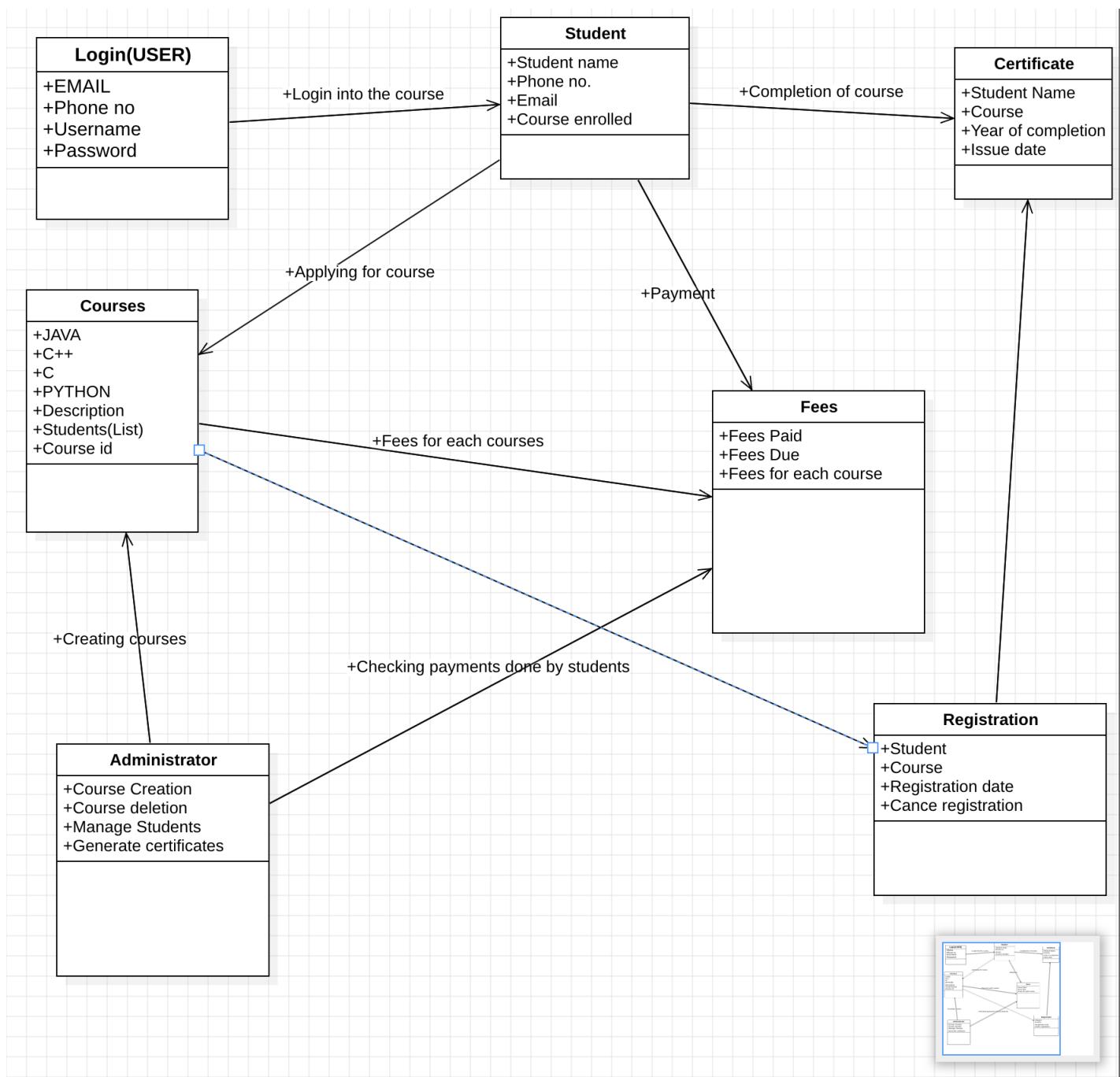
Elements in a Class Diagram:

- **Class:** Represents a blueprint for objects, containing attributes and methods.
- **Attributes:** Characteristics or properties of a class.
- **Methods:** Actions or functions that a class can perform.
- **Relationships:**
 - **Association:** Represents a link between two classes.
 - **Aggregation:** A "whole-part" relationship between two classes, where the part can exist independently of the whole.
 - **Composition:** A stronger form of aggregation where the part cannot exist without the whole.
 - **Inheritance:** A relationship where one class (subclass) inherits the properties of another class (superclass).
 - **Multiplicity:** Defines the number of instances in the relationship between classes.

Example (Class Diagram for an Online Course Reservation System):

- **Classes:**
 - **Student:**
 - Attributes: StudentID, Name, Email.
 - Methods: Register(), ViewCourses(), CancelReservation().
 - **Course:**
 - Attributes: CourseID, CourseName, Instructor, Capacity.
 - Methods: UpdateCourseDetails(), GetAvailableSeats().
 - **Instructor:**
 - Attributes: InstructorID, Name, Email.
 - Methods: AssignCourse(), UpdateCourseDetails().
 - **Admin:**
 - Attributes: AdminID, Name, Email.
 - Methods: AddCourse(), RemoveCourse().
- **Relationships:**
 - **Student registers for a Course** (association).
 - **Instructor manages a Course** (aggregation).
 - **Admin adds/removes a Course** (association).

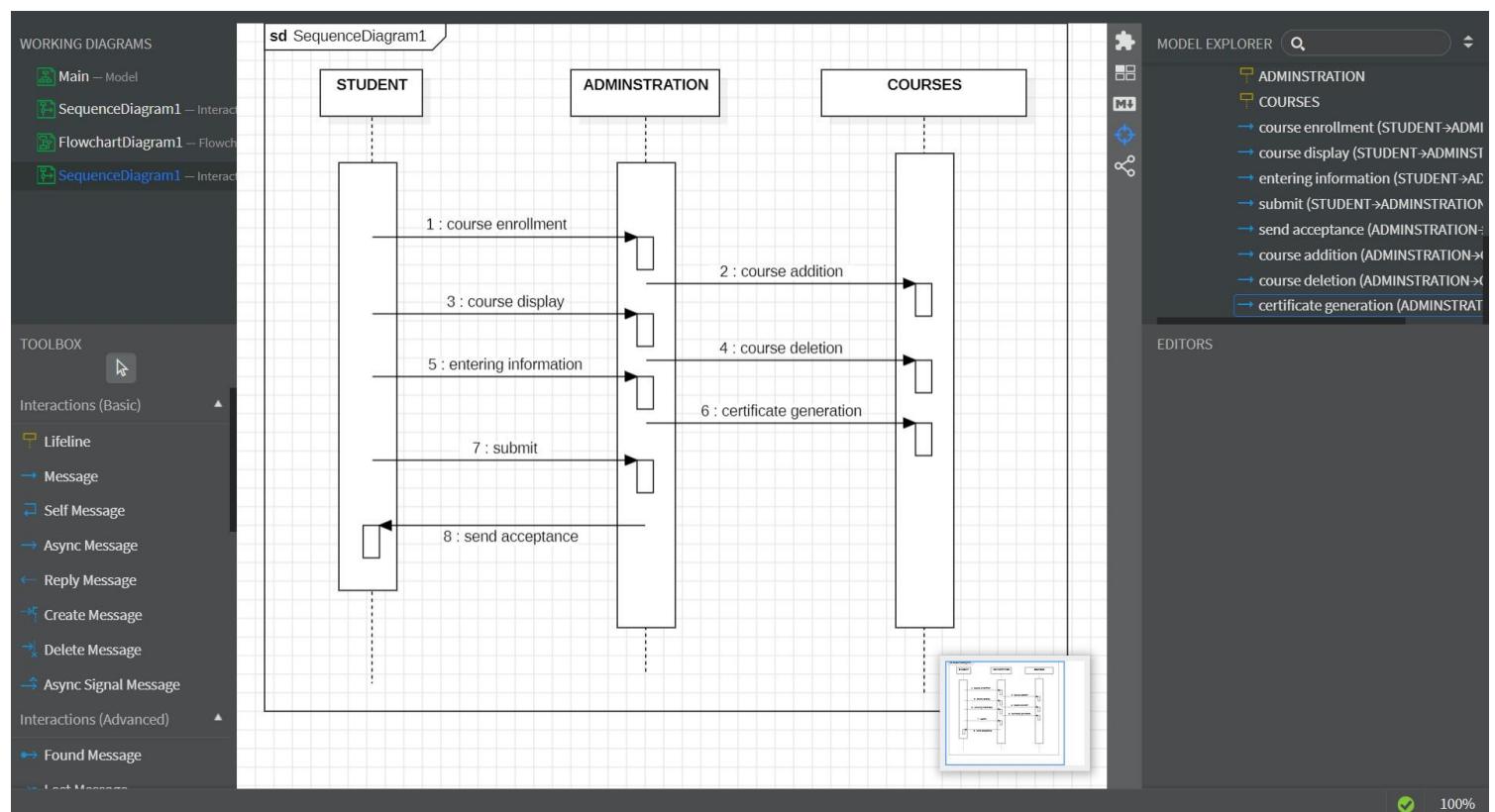
Class Diagram:-



SEQUENCE DIAGRAM:

- It shows interactions between three components: **STUDENT**, **ADMINISTRATION**, and **COURSES**.
- The sequence of actions includes:
 1. **Course enrollment** initiated by the student and sent to administration.
 2. **Course addition** from the administration to the courses component.
 3. **Course display** sent back to the student.
 4. **Course deletion** handled by the administration.
 5. **Entering information** sent by the student to the administration.
 6. **Certificate generation** triggered by the administration in the courses section.
 7. **Submit** action initiated by the student.
 8. **Send acceptance** back from administration to the student.

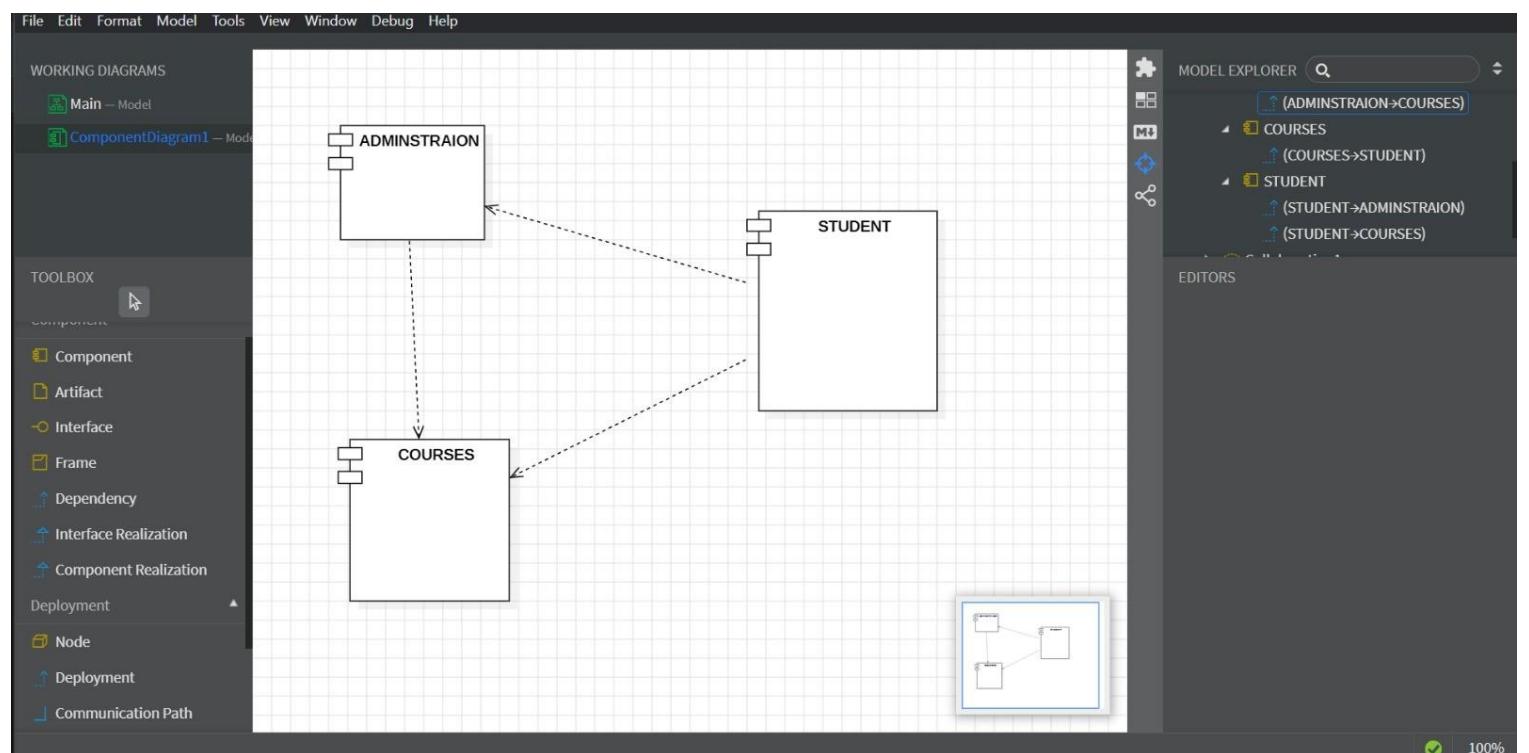
Diagram:-



COMPONENT DIAGRAM:

- It illustrates the relationships between three components: **ADMINISTRATION**, **COURSES**, and **STUDENT**.
- Solid lines represent dependencies or directed associations between these components.
- ADMINISTRATION interacts with both STUDENT and COURSES, while STUDENT and COURSES are also linked.
- The lines connecting them signify communication or dependency relationships.

Diagram:-



Experiment - 3

1. 2.1 Installing Eclipse

To install Eclipse IDE for Enterprise Java Developers (commonly known as Eclipse Enterprise) on your Windows 10 system, follow these steps:

Step 1: Download Eclipse Installer:

->1.1 Go to the Eclipse Downloads page.

The screenshot shows the official Eclipse Downloads page. At the top, there's a navigation bar with links for Home, Downloads, Documentation, and Support. Below it is a large search bar. The main content area features a stylized car icon and the text "Download Eclipse Technology that is right for you". A prominent orange banner at the bottom of the main section promotes the "Build the Car of the Future at Eclipse SDV Hackathon 2024" event, with "Learn More" and "Register" buttons. Below this, two main download sections are shown: one for the "Eclipse IDE" (with a blue logo and "Install your favorite desktop IDE packages" text) and one for "TEMURIN" (with a red logo and "The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem" text). Both sections have "Learn More" and "Download" buttons. At the very bottom of the page, there's a cookie consent banner with "Decline" and "Allow cookies" buttons.

->1.2 Find "Eclipse IDE for Enterprise Java and Web Developers."

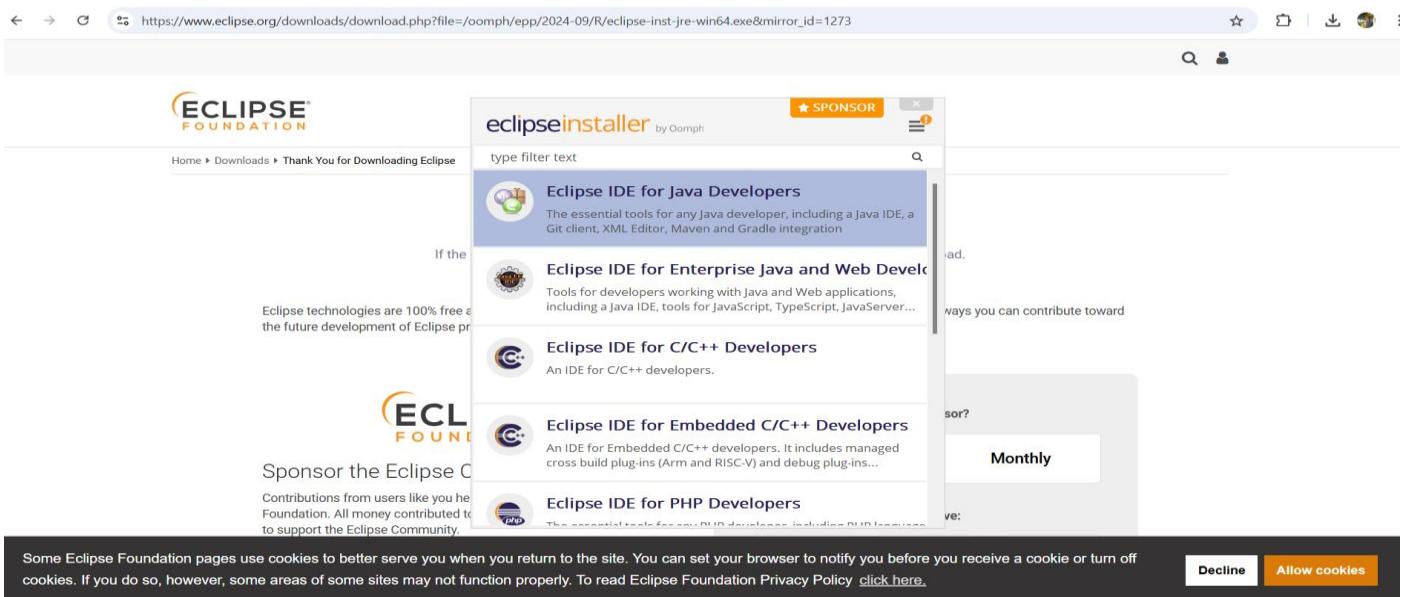
->1.3 Click on "Download x86_64" to download the Windows installer.

The screenshot shows the Eclipse Foundation Downloads page after a download has been completed. The main message is "Thank you for your download!". It includes a link to start the download if it didn't begin automatically. Below this, there's information about contributing to Eclipse projects and a "Sponsor the Eclipse Community" section with options for "One Time" or "Monthly" donations. A sidebar on the right shows the "Recent download history" with an entry for "eclipse-inst-jre-win64.exe". At the bottom, there's a cookie consent banner with "Decline" and "Allow cookies" buttons.

Step 2: Run the Eclipse Installer:

-> 2.1 Locate the downloaded installer (eclipse-inst-jre-win64.exe).

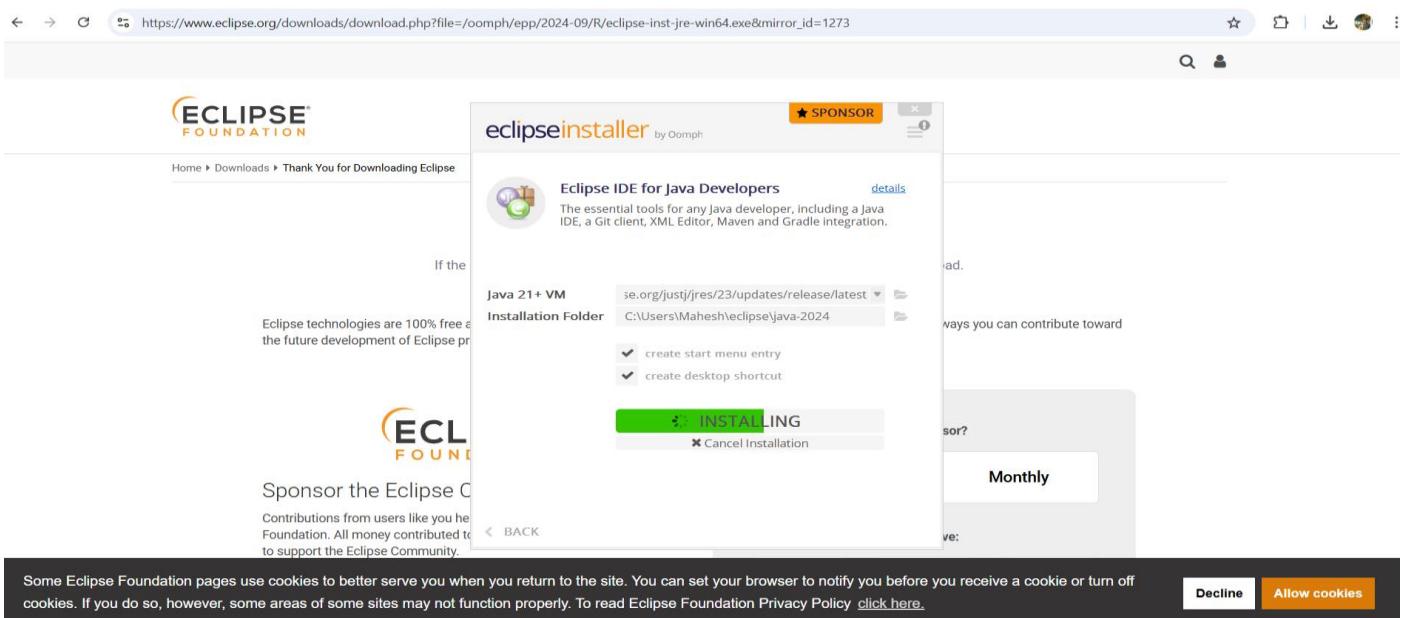
->2.2 Double-click the installer to run it.



->2.3 In the installer, choose:

-> "Eclipse IDE for Enterprise Java and Web Developers."

-> 2.4 Select the installation folder (use default or choose another).



->2.5 Ensure the JDK path is correct (auto-detected or browse manually).

-> 2.6 Click "INSTALL" to begin the installation.

->2.7 Accept the license agreement to proceed.

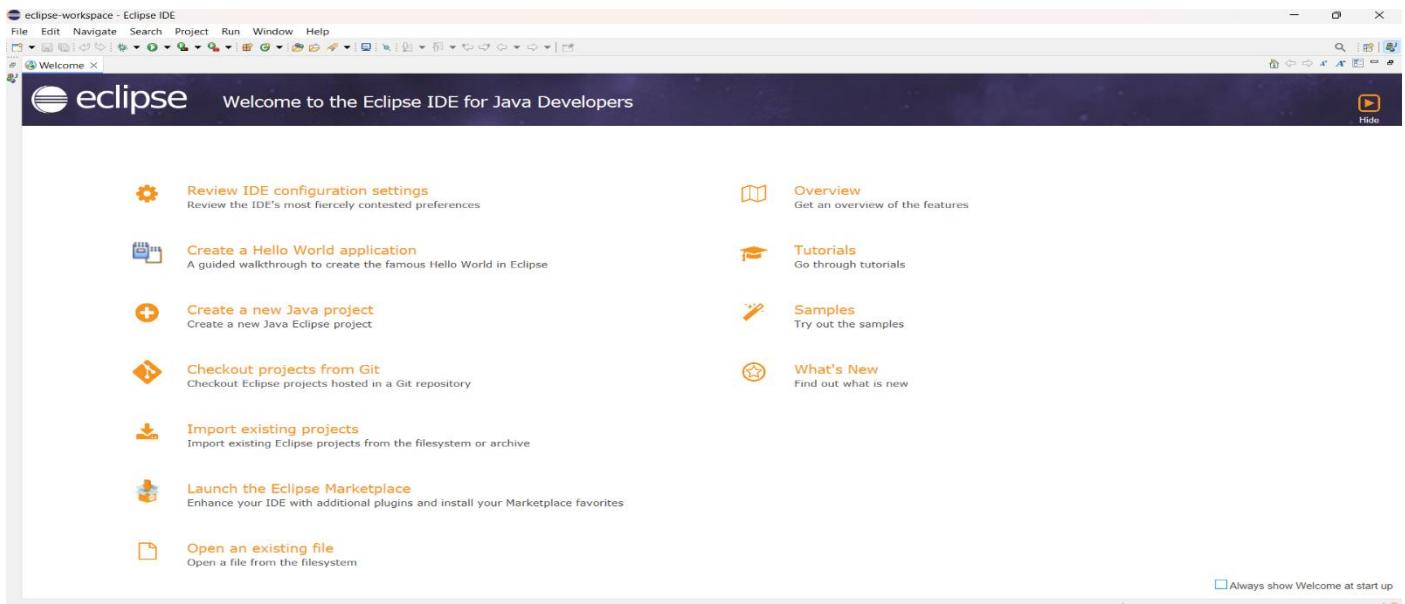
Step 3: Launch Eclipse:

->3.1 After installation, click "LAUNCH" to start Eclipse.

The screenshot shows a web browser displaying the Eclipse Foundation download page at https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2024-09/R/eclipse-inst-jre-win64.exe&mirror_id=1273. A modal window titled "eclipseinstaller by Oomph" is open, prompting the user to select an installation folder (C:\Users\Mahesh\eclipse\java-2024) and choose options like creating a start menu entry or desktop shortcut. The "LAUNCH" button is highlighted. The background shows the Eclipse Foundation logo and a message about the Eclipse IDE for Java Developers.

->3.2 Select a workspace folder for projects and settings (default or custom).

->3.3 Eclipse will start, and you're ready to create or import projects.



2.2. Download Apache Tomcat 9.0

1. Download Apache Tomcat:

-> 1.1 Go to Apache Tomcat 9.0 download page.

-> 1.2 Under "Binary Distributions" -> "Core" section, download the 32-bit/64-bit Windows Service Installer (.exe).

2. Run the Installer:

->2.1 Locate the .exe file and double-click to start the installation.

->2.2 Follow these installation steps:

->Welcome Screen -> Click Next.

-> License Agreement -> Accept and click Next.

-> Choose Components -> Leave defaults and click Next.

->Configuration:

-> Set the HTTP/1.1 Connector Port (default: 8080).

-> Optionally, set Tomcat Administrator login.

->Click Next.

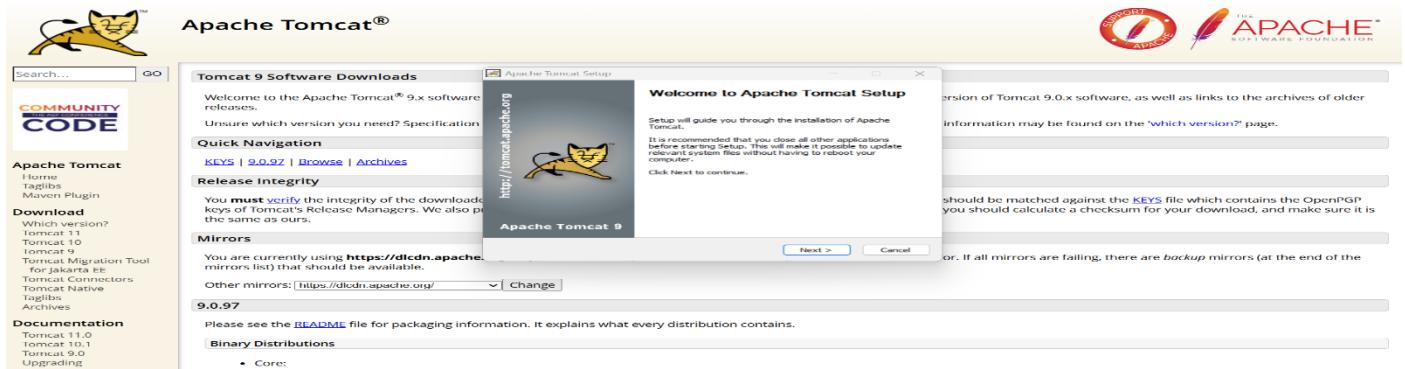
->Choose Java Virtual Machine -> Ensure correct JDK path.

->Install Location -> Choose location or leave default.

->Click Next, then Install.

3. Complete Installation:

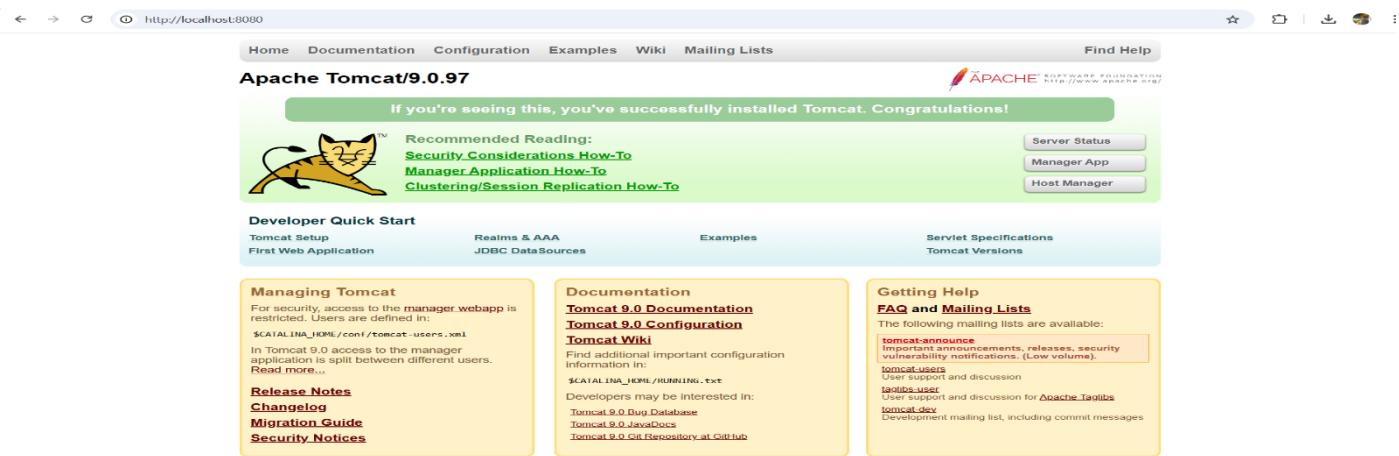
->3.1 After installation, choose to start Tomcat if desired.



-> 3.2 Click Finish to exit the installer.

4. Verify Installation:

->4.1 Open a web browser -> Go to <http://localhost:8080>.



->4.2 Tomcat homepage should display, confirming the installation.

Tomcat Web-Server on Eclipse

1. Ensure You Have:

-> Eclipse IDE for Java EE Developers installed.

-> Apache Tomcat 9.0 installed.

2. Open Eclipse and Configure the Server:

->2.1 Open Eclipse -> Go to Window -> Show View -> Other... -> Servers.

-> 2.2 Add a new server:

- > Right-click in Servers view -> New -> Server.
- >Expand "Apache" -> Select "Tomcat v9.0" -> Click Next.
- > Set Server Name (optional).
- >Browse to Tomcat installation directory (e.g., C:\Program Files\Apache Software Foundation\Tomcat 9.0).
- >Ensure correct JRE is selected -> Click Finish.

3. Add Projects to Server (optional):

- >3.1 In the wizard, add projects if desired or skip this step.

4. Test Server Configuration:

- > 4.1 Start the server:

- >Right-click on the Tomcat server in the Servers view -> Select Start.

- > Console should show Tomcat is running.

- >4.2 Access Tomcat:

- > Open a browser -> Go to http://localhost:8080.

- > Tomcat homepage confirms the server is running correctly.

Experiment – 4A

Git Basics Commands

- **Git version:** check If git is installed using git --version

```
singi@BUNNY MINGW64 ~
$ git --version
git version 2.42.0.windows.2
```

- **Git Config:** Set up your name and email using git config
 - git config --global user.name "Your name"
 - git config --global user.email "Your email"

```
singi@BUNNY MINGW64 ~
$ git config --global user.name "pavansingireddy123"

singi@BUNNY MINGW64 ~
$ git config --global user.email "pavansingireddy3@gmail.com"
```

- **Git init:** start a new project by creating a git repository
 - Example: git init creates an empty git project

```
singi@BUNNY MINGW64 ~ (main)
$ mkdir test

singi@BUNNY MINGW64 ~ (main)
$ cd test

singi@BUNNY MINGW64 ~/test (main)
$ git init
• Initialized empty Git repository in C:/Users/singi/test/.git/
```

- **Git status:** Check the current state of your files and see if any changes need to be saved.
- **Command:** git status

```

Mahesh@DESKTOP-D7E00GU MINGW64 ~ (master)
$ cd proj1
Mahesh@DESKTOP-D7E00GU MINGW64 ~ (master)
$ pwd
/c/Users/Mahesh/proj1
Mahesh@DESKTOP-D7E00GU MINGW64 ~/proj1 (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>
http.sslcert=
http.sslkey=
core.fscache=true
core.symlinks=false
core.ignorecase=true
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
Mahesh@DESKTOP-D7E00GU MINGW64 ~/proj1 (master)
$ git init
Initialized empty Git repository in c:/users/Mahesh/proj1/.git/
Mahesh@DESKTOP-D7E00GU MINGW64 ~/proj1 (master)
$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
Mahesh@DESKTOP-D7E00GU MINGW64 ~/proj1 (master)
$
```

- **Git add:** Stage changes by adding the files to the staging area before saving.
- Example: git add . stages all change

```

singi@BUNNY MINGW64 ~/test (main)
$ notepad f1

singi@BUNNY MINGW64 ~/test (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f1.txt

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

- **Git commit:** Save your changes with a message explaining what was changed.
- Example: git commit -m “Added a new feature”

```

singi@BUNNY MINGW64 ~/test (main)
$ git commit -m "info file added"
[main (root-commit) 33ad2a6] info file added
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 f1.txt
```

- **Git diff:** Compare the changes between your current lines and the last commit.
- **Command:** git diff
- Example: Run git diff to see the differences between your current work and the last saved version.

```

singi@BUNNY MINGW64 ~/test (main)
$ notepad f1.txt

singi@BUNNY MINGW64 ~/test (main)
$ git diff
diff --git a/f1.txt b/f1.txt
index e69de29..95d09f2 100644
--- a/f1.txt
+++ b/f1.txt
@@ -0,0 +1 @@
+hello world
\ No newline at end of file

```

- Git help: Get help on any git command.
- Command: git help <command>
- Example: Use git help add to learn more about the git add command.

```

singi@BUNNY MINGW64 ~/test (main)
$ git help add

```



Experiment – 4B

1. Collaboration:

- **git remote:** Adds a new remote repository. This is useful when you want to link your local repository with a remote one, like GitHub.

git remote add <name><url>

```
singi@BUNNY MINGW64 ~/test (main)
$ git remote add origin https://github.com/pavansingireddy123/sample.git

singi@BUNNY MINGW64 ~/test (main)
$ git remote -v
origin  https://github.com/pavansingireddy123/sample.git (fetch)
origin  https://github.com/pavansingireddy123/sample.git (push)
```

- **git push:** Used to transfer the commits or pushing the content from the local repository to the remote repository. The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

git push -u origin <master|current directory>

```
singi@BUNNY MINGW64 ~/test (main)
$ git remote -v
origin  https://github.com/pavansingireddy123/sample.git (fetch)
origin  https://github.com/pavansingireddy123/sample.git (push)

singi@BUNNY MINGW64 ~/test (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 212 bytes | 212.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/pavansingireddy123/sample.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

- **git clone:** used to create a local working copy of an existing remote repository. The command downloads the remote repository to the local computer.

gitclone <remote url>

```
singi@BUNNY MINGW64 ~/test (main)
$ git clone https://github.com/pavansingireddy123/virat.git
Cloning into 'virat'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
```

- **git fetch :** Downloads changes from a remote repository without applying them to your working directory. You can later merge these changes.

git fetch <remote>

```
singi@BUNNY MINGW64 ~/test (main)
$ git fetch origin main
From https://github.com/pavansingireddy123/sample
 * branch          main       -> FETCH_HEAD
```

- **git pull:** Fetches and integrates changes from a remote repository by rebasing instead of merging. This creates a linear history.

```
git pull <remote><branch>
```

```
singi@BUNNY MINGW64 ~/test (main)
$ git pull
Already up to date.
```

Using SSH Key for Authentication in Git-Github:

Using SSH keys in GitHub allows you to securely authenticate your GitHub account from your computer without needing to repeatedly enter your username and password. This method is more secure and convenient, especially for frequent interactions with GitHub repositories.

Step 1: Set Up SSH

1.1 Check for Existing SSH Keys:

->Open terminal and run:

```
ls -al ~/.ssh`
```

->If you see `id_rsa` and `id_rsa.pub`, you already have an SSH key.

1.2 Generate SSH Key (if not already present):

->Run the following command:

```
'ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

```
singi@BUNNY MINGW64 ~/test (main)
$ ssh-keygen -t rsa -C pavansingireddy3@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/singi/.ssh/id_rsa):
Created directory '/c/Users/singi/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/singi/.ssh/id_rsa
Your public key has been saved in /c/Users/singi/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:iYXtyxtOwp83u/GyjQcHKEyoYtKuepPgkIBBGTZDkqY pavansingireddy3@gmail.com
The key's randomart image is:
+---[RSA 3072]---+
|oBo . .
|++o . .o |
|+. . o. o. |
|E.o o+... |
|o+ ..S . |
|+ . . . . . |
|++ . o = .o |
|+.+ = +++. |
|+. . =.=Bo |
+---[SHA256]---+
```

-> Press Enter to accept the default location.

- > Optionally, set a passphrase for added security.
- > SSH keys are saved in `~/.ssh/id_rsa` (private key) and `~/.ssh/id_rsa.pub` (public key).

Step 2: Add SSH Key to GitHub:

2.1 Copy SSH key to clipboard:

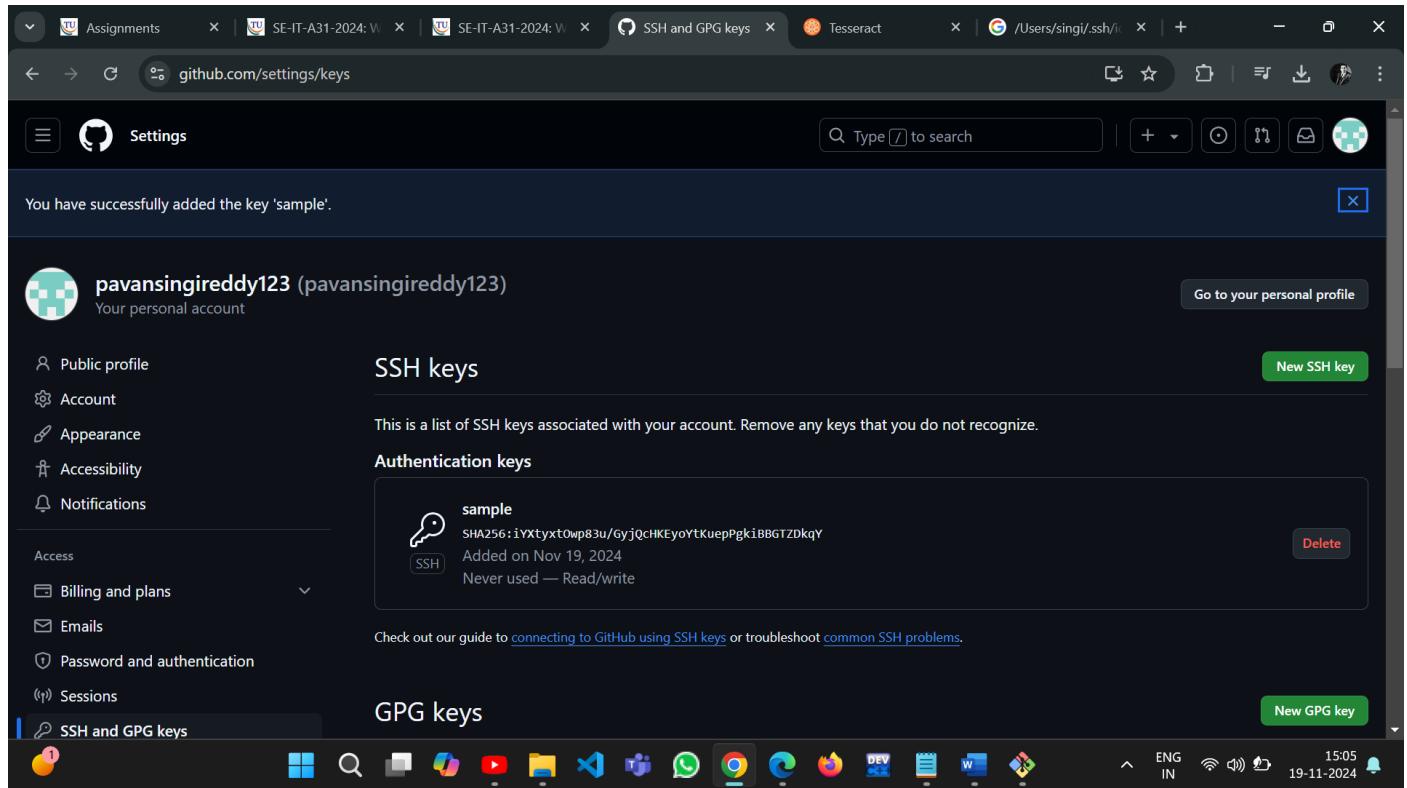
- > Run this command:

```
'cat ~/.ssh/id_rsa.pub'
```

- > Copy the entire output.

2.2 Add SSH Key in GitHub:

- > Go to GitHub -> Settings -> SSH and GPG keys -> New SSH key.



- > Paste the copied key in the "Key" field and give it a title.

- > Click "Add SSH key."

Step 3: Test SSH Connection:

3.1 Test the connection:

- > Run the command:

```
'ssh -T git@github.com'
```

```
singi@BUNNY MINGW64 ~/test (main)
$ ssh -T git@github.com
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOQU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi pavansingireddy123! You've successfully authenticated, but GitHub does not provide shell access.
```

->You should see a message like:

"Hi username! You've successfully authenticated."

Fork :

In Git, a "**fork**" refers to a copy of a repository from one user's account to another user's account on a platform like GitHub. Forking is commonly used to contribute to open-source projects or collaborate on projects where you don't have direct write access to the original repository. Forking allows you to work independently.

Step 1: Fork a Repository:

- > 1.1 Go to the repository you want to fork (e.g., <https://github.com/.../.....>).
- > 1.2 Click the "Fork" button in the top-right corner to create a copy under your account.

Step 2: Clone the Forked Repository:

- > 2.1 Navigate to your forked repository on GitHub.
- > 2.2 Click the "Code" button and copy the HTTPS or SSH URL.
- > 2.3 Clone the repository to your local machine:

```
'git clone git@github.com:yourusername/Spoon-Knife.git'
```

```
singi@BUNNY MINGW64 ~/test (main)
$ git clone https://github.com/pavansingireddy123/Spoon-Knife.git
Cloning into 'Spoon-Knife'...
remote: Enumerating objects: 10, done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 10 (from 1)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (1/1), done.
```

->2.4 Navigate into the repository:

```
'cd Spoon-Knife'
```

```
singi@BUNNY MINGW64 ~/test (main)
$ cd Spoon-Knife
```

Step 3: Make Changes and Push:

->3.1 Create a New Branch:

```
`git checkout -b my-feature-branch`
```

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (main)
$ git checkout -b branch1
Switched to a new branch 'branch1'
```

-> 3.2 Make Changes:

->Edit files or create new ones.

-> Stage changes:

```
`git add .`
```

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git add .
```

->Commit changes:

```
`git commit -m "Add my new feature"'
```

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git commit -m "changes made"
On branch branch1
nothing to commit, working tree clean
```

-> 3.3 Push Changes to GitHub:

```
`git push origin my-feature-branch`
```

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:     https://github.com/pavansingireddy123/Spoon-Knife/pull/new/branch1
remote:
To https://github.com/pavansingireddy123/Spoon-Knife.git
 * [new branch]      branch1 -> branch1
```

Step 4: Pull from Upstream Repository:

-> 4.1 Add Upstream Remote:

```
`git remote add upstream git@github.com:octocat/Spoon-Knife.git'
```

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git remote add upstream git@github.com:pavansingireddy123/Spoon-Knife.git
```

-> 4.2 Fetch and Pull Updates:

-> Fetch latest changes:

```
`git fetch upstream`
```

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git fetch upstream
From github.com:pavansingireddy123/Spoon-Knife
 * [new branch]      branch1    -> upstream/branch1
 * [new branch]      main       -> upstream/main
```

->Merge changes into your branch:

`git pull upstream main`

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git pull upstream main
From github.com:pavansingireddy123/Spoon-Knife
 * branch      main      -> FETCH_HEAD
Already up to date.
```

-> Resolve any conflicts if they arise.

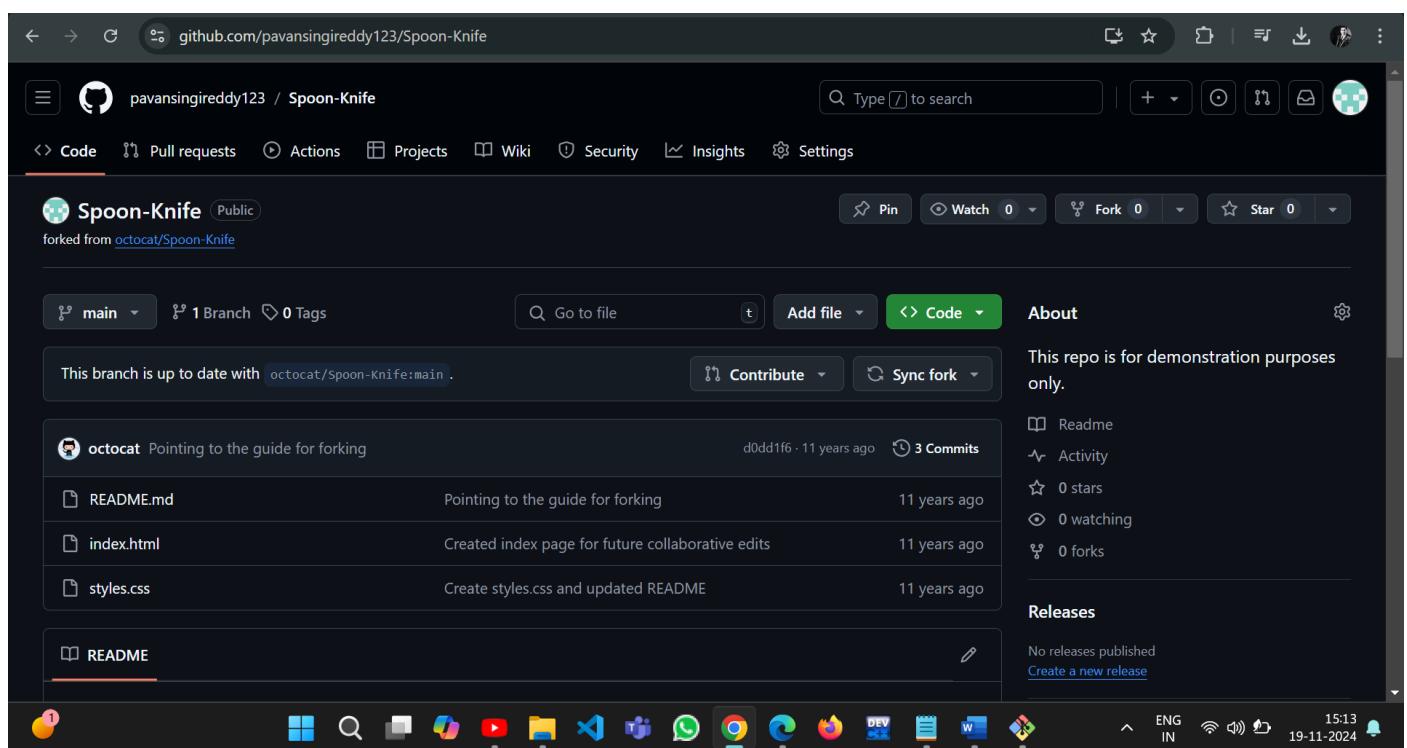
->4.3 Push Merged Changes:

`git push origin my-feature-branch`

```
singi@BUNNY MINGW64 ~/test/Spoon-Knife (branch1)
$ git push origin branch1
Everything up-to-date
```

Step 5: Submit a Pull Request:

->5.1 Go to your forked repository on GitHub.



-> 5.2 Click the banner suggesting to compare and create a pull request.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base repository: octocat/Spoon-Knife base: main ... head repository: penkulamaheshkumar/Spoon-Knife... compare: branch1

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

2 commits 2 files changed 1 contributor

Commits on Nov 16, 2024

files added penkulamaheshkumar committed 6 minutes ago 8ef19f9

changes made penkulamaheshkumar committed 4 minutes ago 2f086ec

Showing 2 changed files with 0 additions and 0 deletions.

Empty file.

sp4

-> 5.3 Review your changes and create the pull request to propose changes to the original repository.

octocat / Spoon-Knife

Code Issues 2.2k Pull requests 5k+ Actions Projects Wiki Security Insights

Branch1 #34427

Open penkulamaheshkumar wants to merge 2 commits into octocat:main from penkulamaheshkumar:branch1

Conversation 0 Commits 2 Checks 0 Files changed 2

penkulamaheshkumar commented now

new files added

penkulamaheshkumar added 2 commits 7 minutes ago

files added 8ef19f9

changes made 2f086ec

This branch has no conflicts with the base branch Only those with write access to this repository can merge pull requests.

Add a comment

Reviewers No reviews Still in progress? Convert to draft

Assignees No one assigned

Labels None yet

Projects None yet

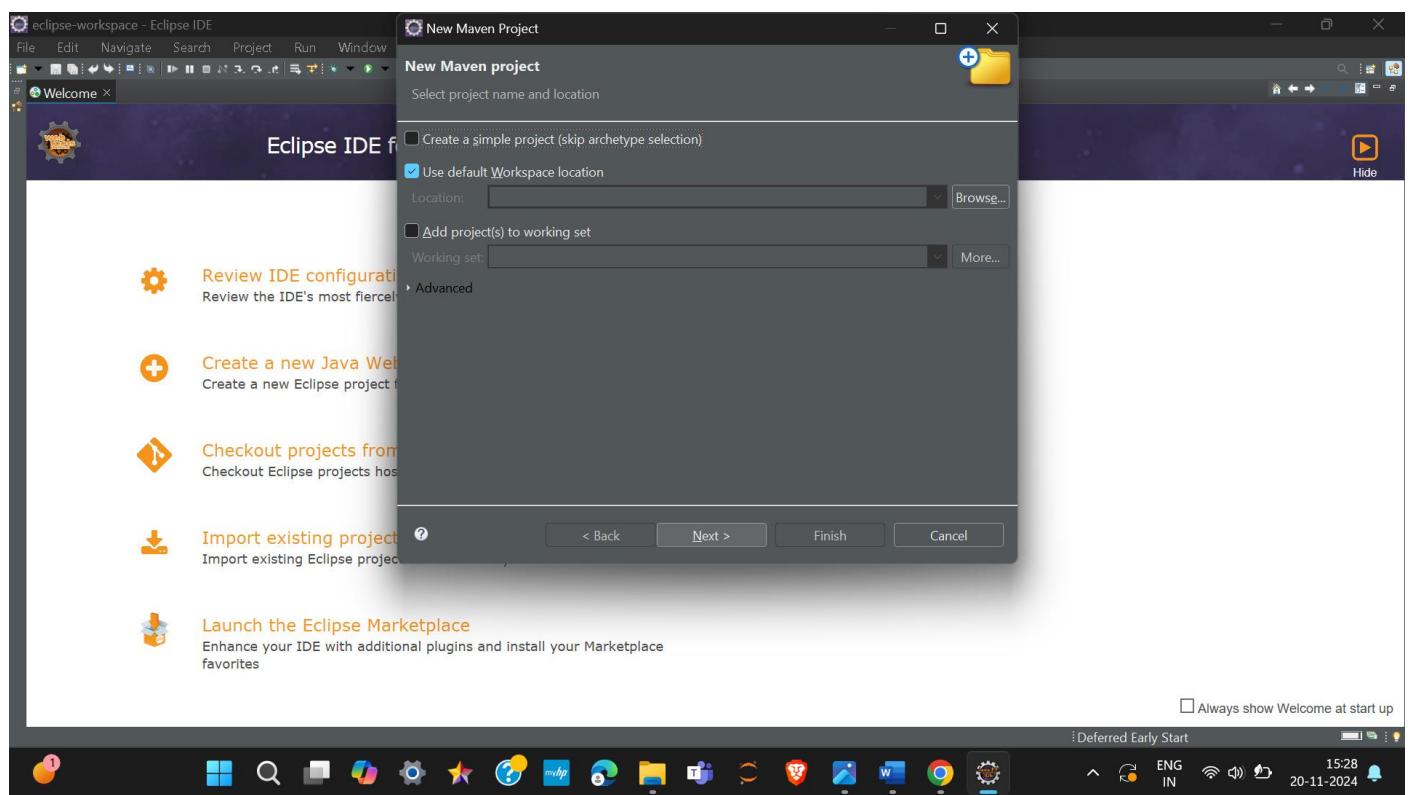
Milestone No milestone

Experiment – 5A

Creation of Maven Java Project

Step 1. Open Eclipse IDE

-> Launch Eclipse workspace



Step 2. Install Maven Plugin (if not installed)

->2.1. Go to "Help" in the top menu

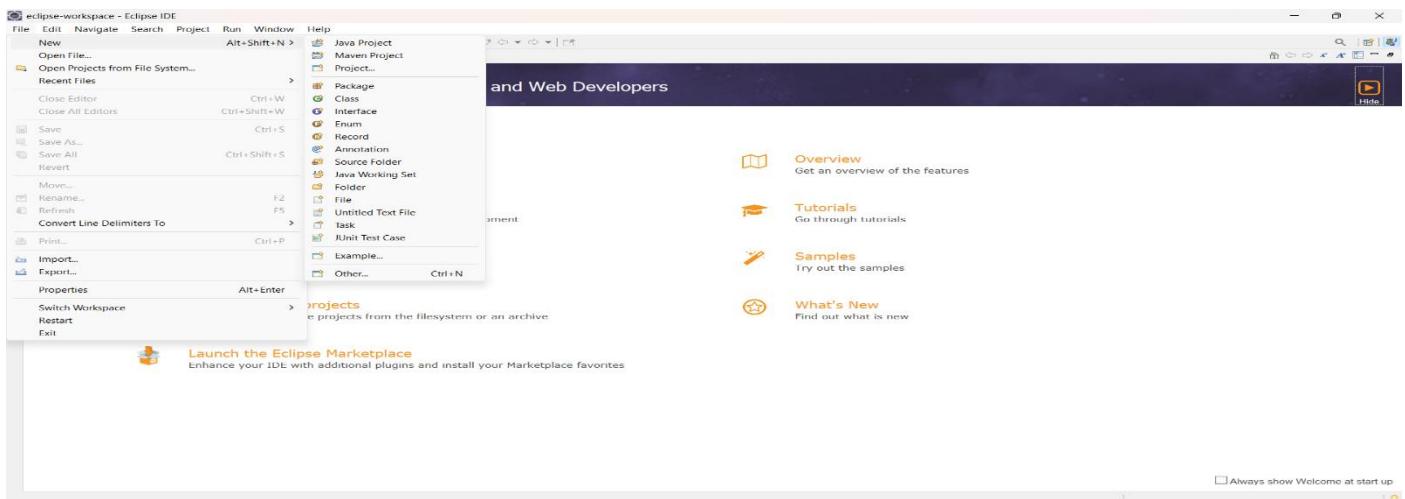
-> 2.1.1. Click "Eclipse Marketplace"

->2.1.2. Search for "Maven Integration for Eclipse"

->2.1.3. Install the plugin if not already installed

Step 3. Create a New Maven Project

->3.1. File -> New -> Project...



->3.1.1. Expand "Maven"

->3.1.2. Select "Maven Project" and click "Next"

Step 4. Set Project Configuration

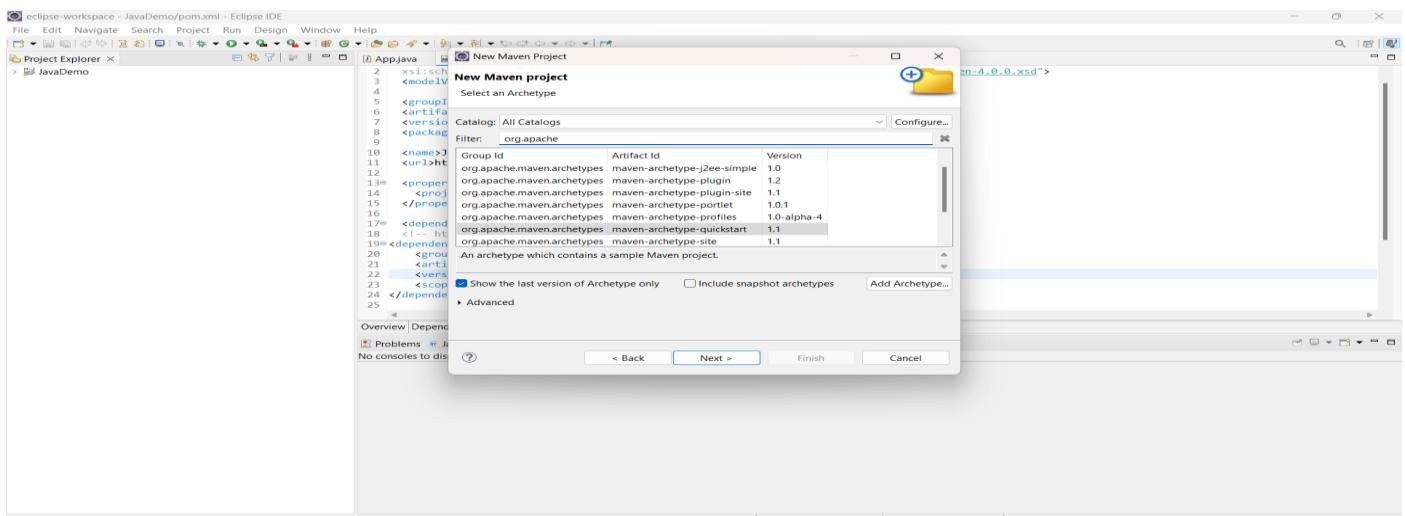
->4.1. Select workspace location (default or custom)

->4.2. Click "Next"

Step 5. Choose Maven Archetype

->5.1. Select an archetype(e.g "org.apache.maven.archetypes -> maven-archetype-quickstart 1.4 ")

->5.2. Click "Next"



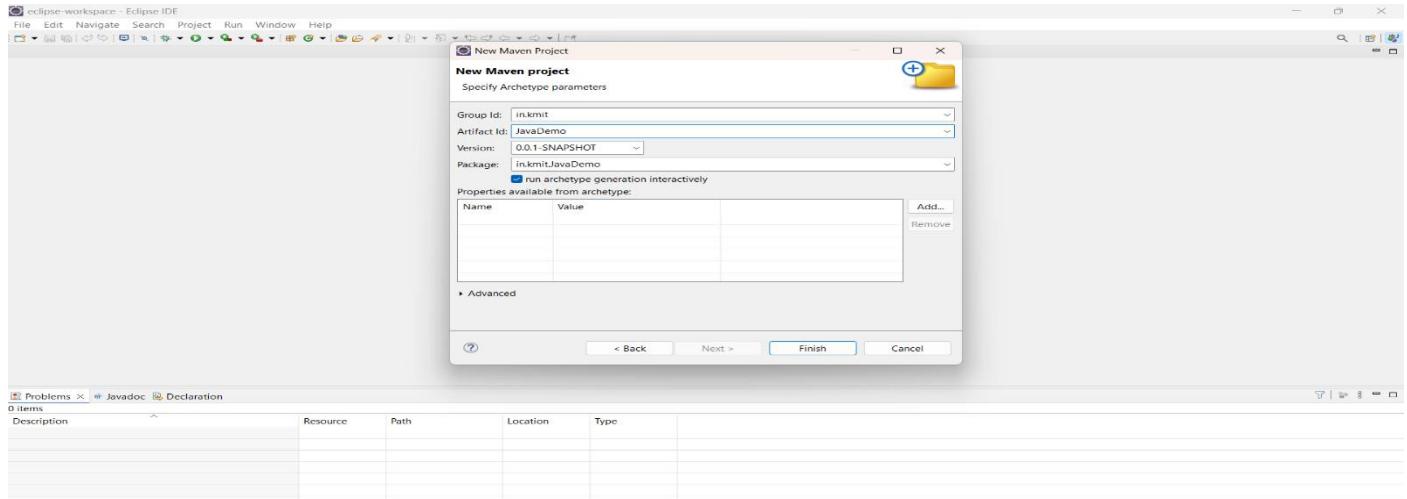
Step 6. Define Project Metadata

-> 6.1. Group ID: (e.g., com.example)

->6.2. Artifact ID: (e.g., my-maven-project)

->6.3. Version: (default is usually fine)

->6.4. Click "Finish"



In Console, artifacts are grouped. When prompted with Y/N, type 'Y'.

Step 7. Maven Project Created

->7.1. Project structure is generated with a standard Maven layout

->7.2. Includes:

-> src/main/java (for Java source code)

-> src/test/java (for test code)

->pom.xml (Maven configuration file)

Step 8. Update Project Settings (if needed)

->8.1. Right-click on the project -> Maven -> Update Project...

-> 8.2. Ensure dependencies are up to date

Step 9. Build and Run Maven Project

->9.1. Right-click on App.java -> Run As -> Maven Clean

->9.1.1. Right-click on App.java -> Run As -> Maven Install

->9.1.2. Right-click on App.java -> Run As -> Maven Test

->9.1.3. Right-click on App.java -> Run As -> Maven Build

The screenshot shows the Eclipse IDE interface. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The left sidebar has a 'File' menu with options like New, Open File..., Save, and Exit. The central workspace shows a Java file named App.java with the following code:

```

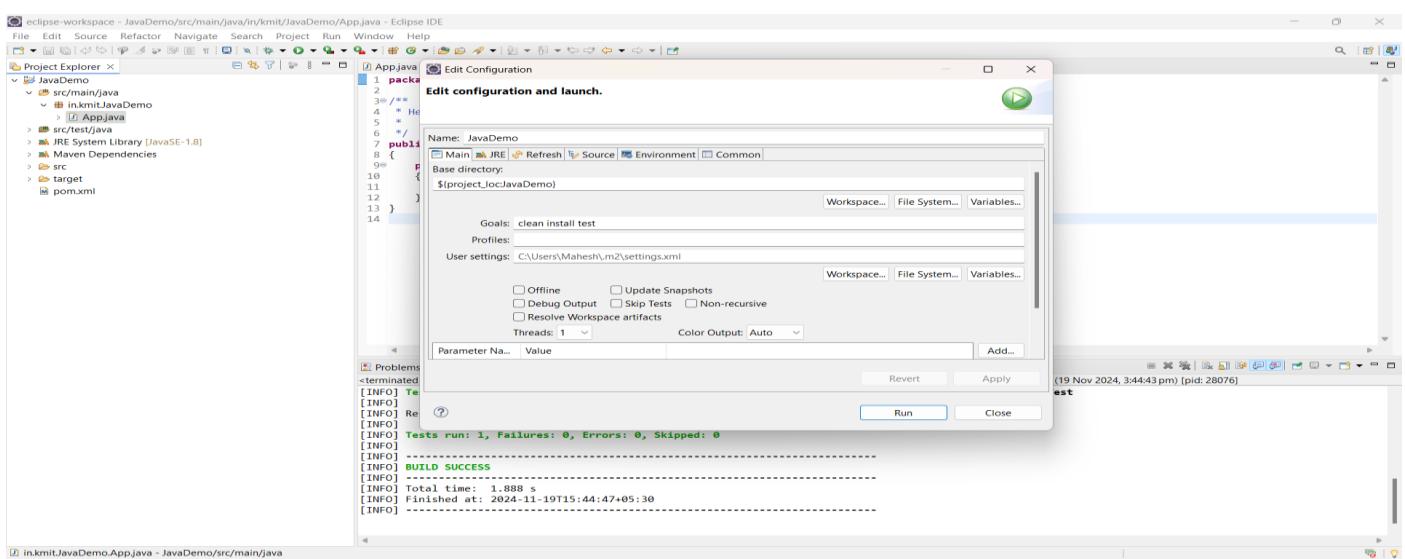
public class App {
    public static void main( String[] args ) {
        System.out.println( "Hello World!" );
    }
}

```

The right side shows the 'Outline' view with the class 'App' and its main method. Below the workspace is a 'Console' tab showing the output: <terminated> App [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (22-Nov-2024, 10:29:59 am – 10:30:00 am) [pid: 24700] Hello World!

Step 10. In the Maven Build dialog:

- >Enter Goals: clean install test
- > Click on Apply -> Click on Run



Step 11. Check console for BUILD SUCCESS message.

Step 12. Run the application:

- >Right-click on App.java -> Run As -> Java Application

->Output: "Hello World" displayed.

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

- Project Explorer:** Shows a project named "java1" containing a package "in.kmit.java1" with an "App.java" file.
- Code Editor:** Displays the "App.java" code:

```
1 package in.kmit.java1;
2
3 /**
4 * Hello world!
5 */
6
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12    }
13 }
```
- Console:** Shows the output of the application:

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (22-Nov-2024, 10:29:59 am – 10:30:00 am) [pid: 24700]
Hello World!
```
- Outline:** Shows the class structure with the main method highlighted.

Experiment – 5B

Creation of Maven web Java Project

Step 1: Open Eclipse

->1.1 Launch Eclipse IDE.

->1.2 Select or create a workspace.

Step 2: Create a New Maven Project

->2.1. File -> New -> Project...

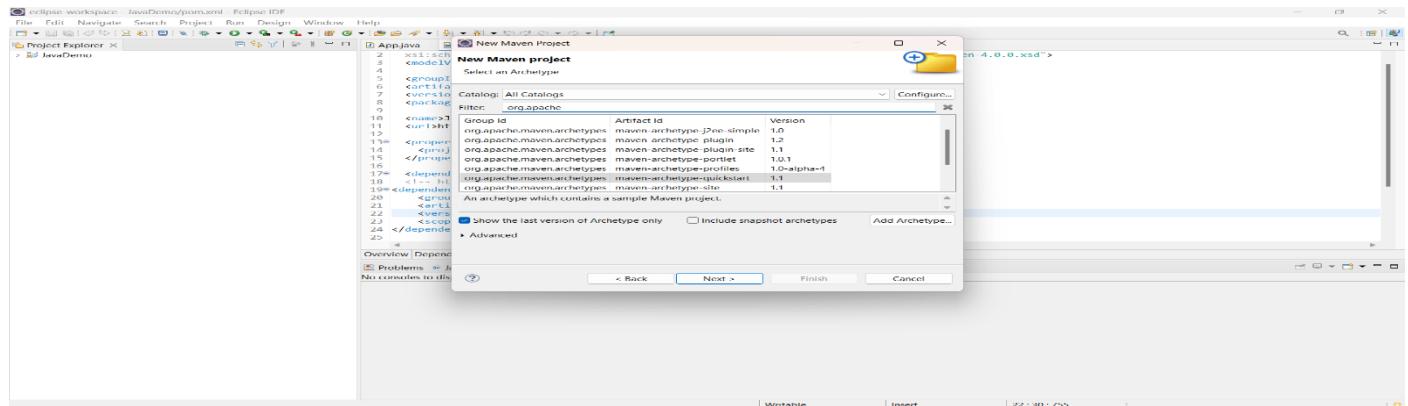
-> 2.1.1. Expand "Maven"

->2.1.2. Select "Maven Project" and click "Next"

Step 3: Choose Maven Archetype

->3.1. Select an archetype(e.g "'org.apache.maven.archetypes' -> 'maven-archetype-webapp' 1.4 ")

->3.2. Click "Next"

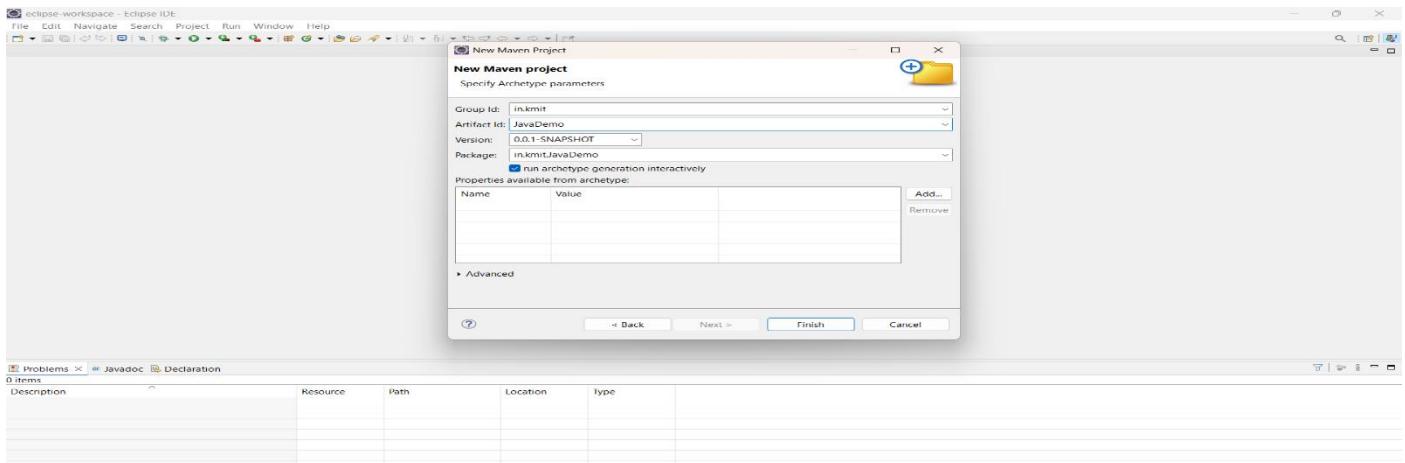


Step 4: Configure the Maven Project

->4.1 Group Id: Enter a group ID (e.g., com.example).

->4.2 Artifact Id: Enter an artifact ID (e.g., my-web-app).

->4.3 Click **Finish** to create the project.



Step 5: Add Maven Dependencies

- > 5.1 Open the **pom.xml** file in the Maven project.
- > 5.2 Add the necessary dependencies for your web project (e.g., Servlet, JSP):

Go to browser -> Open mvnrepository.com

Search for 'Java Servlet API' -> Select the latest version.

Copy the dependency code -> Paste it in MavenWeb's pom.xml under the target folder

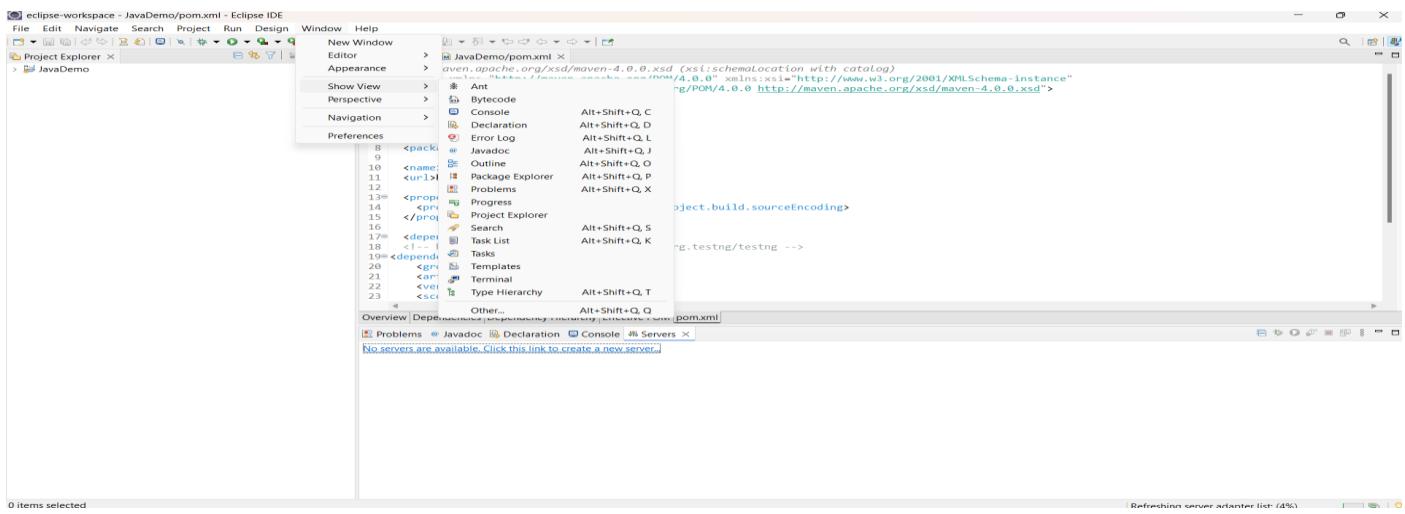
Example:

```
```xml
<dependency>
 <groupId>javax.servlet</groupId>
 <artifactId>javax.servlet-api</artifactId>
 <version>4.0.1</version>
 <scope>provided</scope>
</dependency>
```
```

```

## Step 6:- Configure server:

- >Window -> Show View -> Servers
- >Add server -> Select Tomcat v9.0 server -> Click Next
- >Configure server options (e.g., ports, server location).



## Step 7:- Modify 'tomcat-users.xml':

->Add role and user details under <tomcat-users> tag.

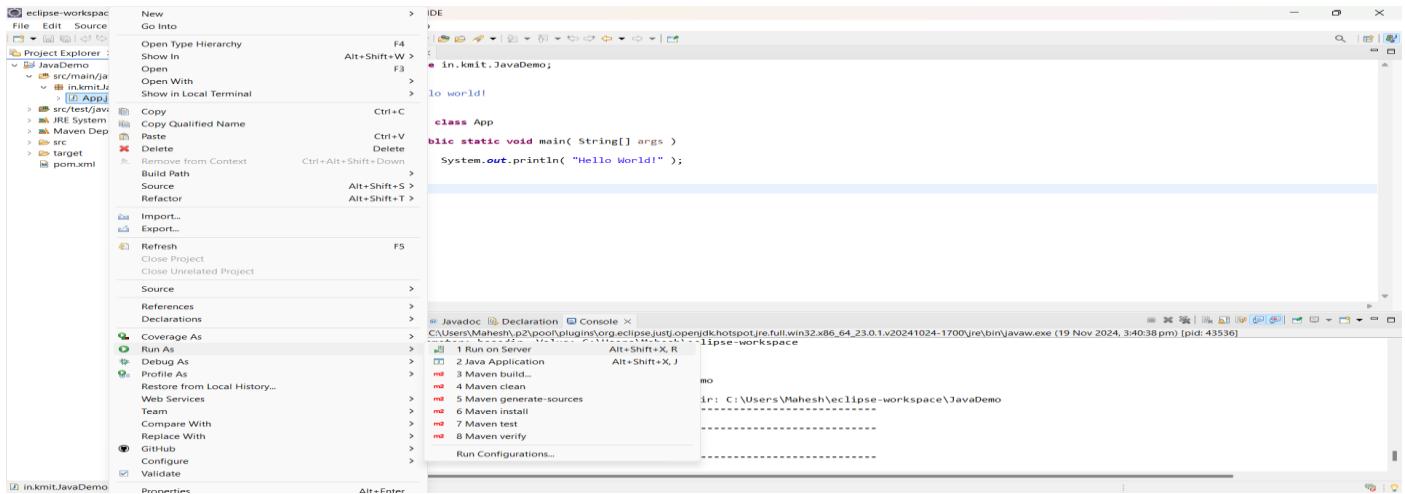
## Step 8.. Build the project:

-> Right-click on index.jsp -> Run As -> Maven Clean

->Right-click on index.jsp -> Run As -> Maven Install

->Right-click on index.jsp -> Run As -> Maven Test

->Right-click on index.jsp -> Run As -> Maven Build



## Step 9. In the Maven Build dialog:

-> Enter Goals: clean install test

->Click on Apply -> Click on Run

eclipse-workspace - JavaDemo/src/main/java/in/kmit/JavaDemo - Eclipse IDE

File Edit Source Refactor Navigate Project Run Window Help

Project Explorer X

JavaDemo

src/main/java

in.kmit.JavaDemo

App.java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

pom.xml

App.java

```
1 package in.kmit.JavaDemo;
2
3 /**
4 * Hello world!
5 */
6
7 public class App {
8
9 public static void main(String[] args)
10 {
11 System.out.println("Hello World!");
12 }
13 }
```

Edit Configuration

Edit configuration and launch.

Name: JavaDemo

Main JRE Refresh Source Environment Common

Base directory: \$project\_loc:JavaDemo

Goals: clean install test

Profiles:

User settings: C:\Users\Mahesh\Unz\settings.xml

Threads: 1

Color Output: Auto

Parameter Name Value Add... Revert Apply

(19 Nov 2024, 3:44:43 pm) [pid: 28076]

est

[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----

[INFO] Total time: 1,888 s

[INFO] Finished at: 2024-11-19T15:44:47+05:30

[INFO] -----

Step 10. Check console for BUILD SUCCESS message.

Step 11. Run the application:

->Right-click on index.jsp -> Run As -> Run on Server

->Select the Tomcat server -> Click on Finish

Step 12. Output: "Hello World" webpage displayed.

eclipse-workspace - JavaDemo/src/main/java/in/kmit/JavaDemo - Eclipse IDE

File Edit Source Refactor Navigate Project Run Window Help

Project Explorer X

JavaDemo

src/main/java

in.kmit.JavaDemo

App.java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

pom.xml

App.java

```
1 package in.kmit.JavaDemo;
2
3 /**
4 * Hello world!
5 */
6
7 public class App {
8
9 public static void main(String[] args)
10 {
11 System.out.println("Hello World!");
12 }
13 }
```

Console

<terminated> App Java Application C:\Users\Mahesh\p2\pool\plugins\org.eclipse.jst.j2ee.core\2.20.0.v20241024-1700\re\bin\javaw.exe (19 Nov 2024, 3:45:49 pm - 3:45:49 pm) [pid: 399]

Hello World!

# Experiment - 6A

## Jenkins Automation

Steps for MavenJava Automation:

Step 1: Open Jenkins (localhost:8080)

Click on "New Item" (left side menu)

The screenshot shows the Jenkins dashboard with the following details:

- Left Sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". It also shows a "Build Queue" section stating "No builds in the queue." and a "Build Executor Status" section showing "0/2".
- Top Bar:** Includes a search bar, help icon, user info for "Singi Reddy Pavan Kumar Reddy", and a "log out" button.
- Main Content:** A table listing Jenkins projects. The columns are: S (Status), W (Icon), Name (sorted by name), Last Success, Last Failure, and Last Duration. Projects listed include:
  - declarative\_script: Last success 4 days 0 hr, last failure 4 days 0 hr, duration 38 sec.
  - demo: Last success 6 days 22 hr, last failure N/A, duration 0.35 sec.
  - git demo: Last success N/A, last failure N/A, duration N/A.
  - hello\_world\_script: Last success 6 days 18 hr, last failure N/A, duration 6.3 sec.
  - maven\_build: Last success 6 days 18 hr, last failure 6 days 22 hr, duration 25 sec.
  - maven\_test: Last success 6 days 18 hr, last failure 6 days 22 hr, duration 4.7 sec.
  - maven\_web\_build: Last success 6 days 18 hr, last failure N/A, duration 22 sec.

Step 2: Create Freestyle Project (e.g., MavenJava\_Build)

Enter project name (e.g., MavenJava\_Build)

Click "OK"

## New Item

Enter an item name

MavenJava\_Build

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

## Configure the project:

**Description:** "Java Build demo"**Source Code Management:**

Git repository URL: [GitMavenJava repo URL]

## Configure

**General****Source Code Management**

## Build Triggers

## Build Environment

## Build Steps

## Post-build Actions

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

https://github.com/pavansingireddy123/maven\_java.git

Credentials [?](#)

- none -

+ Add

Advanced [▼](#)

Add Repository

**Branches to build:** \*/Main or \*/master

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/main

Add Branch

Repository browser ?

(Auto)

▼

Additional Behaviours

Add ▼

## Build Steps:

**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: clean

**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: install

## 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 ▼

## Post-build Actions:

Add Post Build Action -> "Archive the artifacts"

Files to archive: \*\*/\*

Add Post Build Action -> "Build other projects"

Projects to build: MavenJava\_Test

Trigger: Only if build is stable

Apply and Save

#### Post-build Actions

The screenshot shows the Jenkins post-build actions configuration page. It includes sections for 'Archive the artifacts' (with a pattern '\*\*/\*'), 'Build other projects' (with 'MavenJava\_Test' selected and a note about a misspelling), and an 'Add post-build action' dropdown. Buttons for 'Save' and 'Apply' are at the bottom.

Archive the artifacts  
Files to archive: \*\*/\*  
Advanced

Build other projects  
Projects to build: MavenJava\_Test  
① No such project 'MavenJava\_Test'. Did you mean 'MavenJava\_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

## Step 3: Create Freestyle Project (e.g., MavenJava\_Test)

Enter project name (e.g., MavenJava\_Test)

Click "OK"

The screenshot shows the Jenkins 'New Item' creation interface. It has a search bar, user information, and a breadcrumb navigation (Dashboard > All > New Item). The main area is titled 'New Item' and asks for an item name ('MavenJava\_Test'). It lists three project types: 'Freestyle project' (selected), 'Maven project', and 'Pipeline'. A blue 'OK' button is at the bottom.

Jenkins

Search (CTRL+K)

Singi Reddy Pavan Kumar Reddy log out

Dashboard > All > New Item

New Item

Enter an item name

MavenJava\_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.

OK

## Configure the project:

Description: "Test demo"

## Build Environment:

Check: "Delete the workspace before build starts"

### Add Build Step -> "Copy artifacts from another project"

Project name: MavenJava\_Build

Build: Stable build only // tick at this

Artifacts to copy: \*\*/\*

#### Build Steps

The screenshot shows the configuration for a 'Copy artifacts from another project' build step. It includes fields for 'Project name' (MavenJava\_Build), 'Which build' (Latest successful build), and 'Artifacts to copy' (\*\*/\*). A checkbox for 'Stable build only' is checked.

**Copy artifacts from another project**

Project name ?  
MavenJava\_Build

Which build ?  
Latest successful build

Stable build only

Artifacts to copy ?  
\*\*/\*

Artifacts not to copy ?

### Add Build Step -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: test

Post-build Actions:

### Add Post Build Action -> "Archive the artifacts"

Files to archive: \*\*/\*

Apply and Save

The screenshot shows the configuration for 'Invoke top-level Maven targets' and 'Archive the artifacts' post-build actions. For 'Invoke top-level Maven targets', it specifies MAVEN\_HOME as the Maven Version and test as the Goals. For 'Archive the artifacts', it specifies files to archive as \*\*/\*.

**Invoke top-level Maven targets**

Maven Version  
MAVEN\_HOME

Goals  
test

Advanced ▾

Add build step ▾

**Post-build Actions**

**Archive the artifacts**

Files to archive ?  
\*\*/\*

Save Apply

## Step 4: Create Pipeline View for Maven Java project

Click "+" beside "All" on the dashboard

Enter name: MavenJava\_Pipeline

Select "Build pipeline view" // tick here

Dashboard > All > New Item

## New Item

Enter an item name

MavenBuid\_Pipeline

Select an item type



### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



### Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



### Multi-configuration project

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

## create

### Pipeline Flow:

**Layout:** Based on upstream/downstream relationship

Initial job: MavenJava\_Build

Apply and Save OK

### Step 5: Run the Pipeline and Check Output

Click on the trigger to run the pipeline

Click on the small black box to open the console to check if the build is success



Jenkins

Search (CTRL+K)



Singi Reddy Pavan Kumar Reddy

log out

Dashboard &gt; java\_pipeline &gt;

## Build Pipeline

Run History Configure + Delete Manage

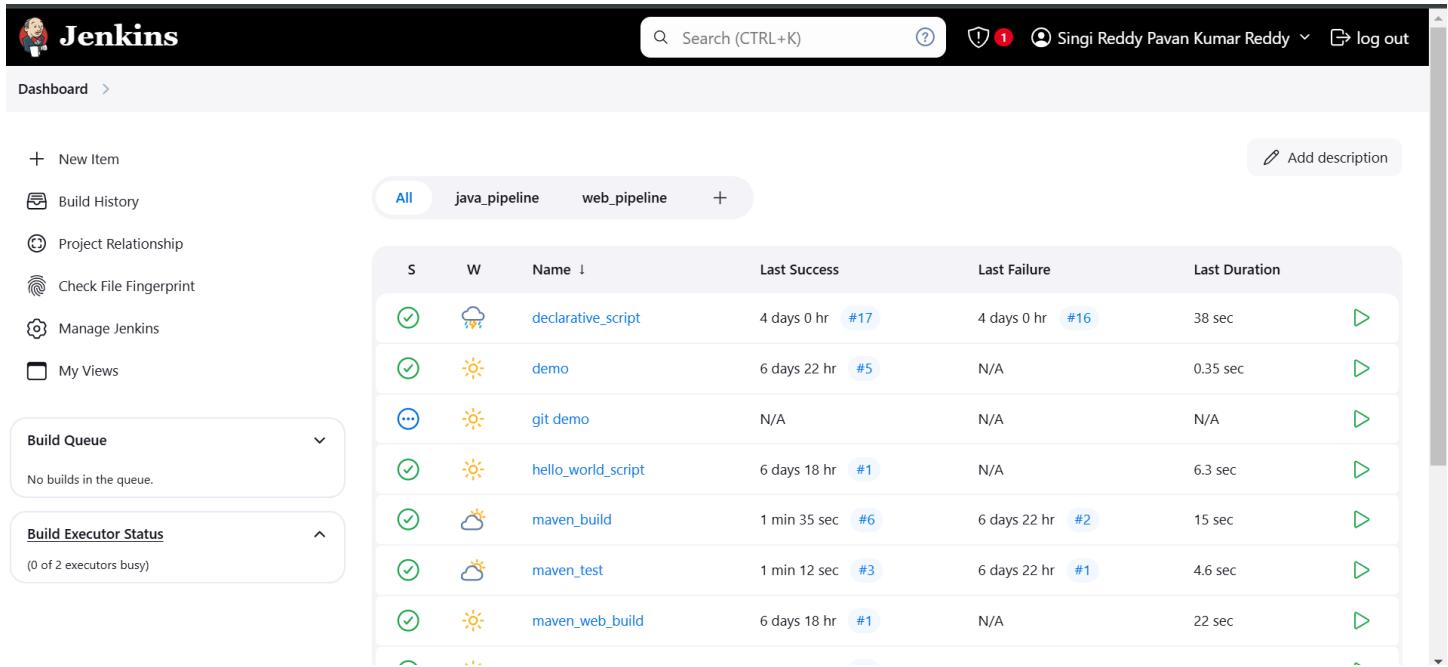


# Experiment – 6B

## Maven Web Automation Steps:

### Step 1: Open Jenkins (localhost:8080)

Click on "New Item" (left side menu)



The screenshot shows the Jenkins dashboard with the following details:

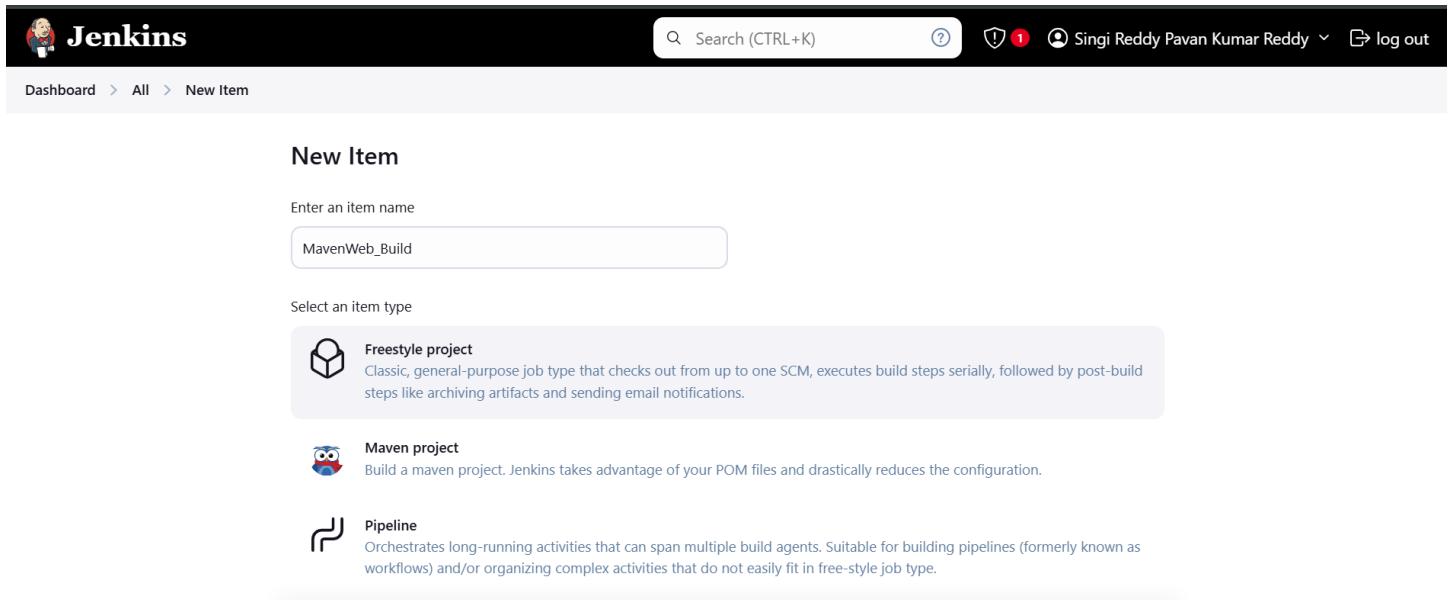
- Left sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". It also displays the "Build Queue" (No builds in the queue) and "Build Executor Status" (0 of 2 executors busy).
- Top navigation:** Shows the Jenkins logo, user name "Singi Reddy Pavan Kumar Reddy", and a "log out" link.
- Search bar:** A search bar with placeholder text "Search (CTRL+K)" and a help icon.
- Job List:** A table listing seven build jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☁️	declarative_script	4 days 0 hr #17	4 days 0 hr #16	38 sec
✓	☀️	demo	6 days 22 hr #5	N/A	0.35 sec
...	☀️	git demo	N/A	N/A	N/A
✓	☀️	hello_world_script	6 days 18 hr #1	N/A	6.3 sec
✓	☁️	maven_build	1 min 35 sec #6	6 days 22 hr #2	15 sec
✓	☁️	maven_test	1 min 12 sec #3	6 days 22 hr #1	4.6 sec
✓	☀️	maven_web_build	6 days 18 hr #1	N/A	22 sec

### Step 2: Create Freestyle Project (e.g., MavenWeb\_Build)

Enter project name (e.g., MavenWeb\_Build)

Click "OK"



The screenshot shows the "New Item" creation page with the following steps:

- Item Name:** The "Enter an item name" field contains "MavenWeb\_Build".
- Item Type:** The "Select an item type" section shows three options:
  - Freestyle project:** Described as a "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." It has a "Freestyle project" icon.
  - Maven project:** Described as "Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration." It has a "Maven project" icon.
  - Pipeline:** Described as "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." It has a "Pipeline" icon.

## Configure the project:

**Description:** "Web Build demo"

### Source Code Management:

Git repository URL: [GitMavenWeb repo URL]

Branches to build: \*/Main or master

### Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#) ✖

Credentials [?](#)

- none - ▼

[+ Add](#)

[Advanced ▼](#)

### Build Steps:

**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: clean

**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: install

## Build Steps

The screenshot shows two separate build step configurations. Both steps are titled "Invoke top-level Maven targets".  
Step 1: Maven Version is set to MAVEN\_HOME, Goals are set to "clean", and the "Advanced" button is visible.  
Step 2: Maven Version is set to MAVEN\_HOME, Goals are set to "install", and the "Advanced" button is visible.

## Post-build Actions:

**Add Post Build Action -> "Archive the artifacts"**

Files to archive: \*\*/\*

**Add Post Build Action -> "Build other projects"**

Projects to build: MavenWeb\_Test

Trigger: Only if build is stable

Apply and Save

The screenshot shows the "Post-build Actions" configuration section with two actions defined:  
1. "Archive the artifacts": Files to archive are set to \*\*/\*.  
2. "Build other projects": Projects to build are set to MavenWeb\_Test, and the trigger is set to "Trigger only if build is stable".  
At the bottom, there are "Save" and "Apply" buttons.

## Step 3: Create Freestyle Project (e.g., MavenWeb\_Test)

Enter project name (e.g., MavenWeb\_Test)

Click "OK"

## Configure the project:

**Description:** "Test demo"

**Build Environment:**

Check: "Delete the workspace before build starts"

**Add Build Step ->** "Copy artifacts from another project"

Project name: MavenWeb\_Build

Build: Stable build only

Artifacts to copy: \*\*/\*

**Build Steps**

**Copy artifacts from another project**

Project name ?  
MavenWeb\_Build

Which build ?  
Latest successful build

Stable build only

Artifacts to copy ?  
\*\*/\*

Artifacts not to copy ?

**Add Build Step ->** "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: test

**Invoke top-level Maven targets** ?

Maven Version  
MAVEN\_HOME

Goals  
test

Advanced ▾

**Post-build Actions:**

**Add Post Build Action ->** "Archive the artifacts"

Files to archive: \*\*/\*

**Add Post Build Action ->** "Build other projects"

Projects to build: MavenWeb\_Deploy

## Apply and Save

### Post-build Actions

Archive the artifacts ? X

Files to archive ?

\*\*/\*

Advanced ▾

Build other projects ? X

Projects to build

MavenWeb\_Deploy

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

## Step 4: Create Freestyle Project (e.g., MavenWeb\_Deploy)

Enter project name (e.g., MavenWeb\_Deploy)

Click "OK"

 Jenkins Search (CTRL+K) ? ! Singi Reddy Pavan Kumar Reddy

Dashboard > All > New Item

### New Item

Enter an item name

MavenWeb\_Deploy

Select an item type

 **Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 **Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 **Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

## Configure the project:

Description: "Web Code Deployment"

### Build Environment:

Check: "Delete the workspace before build starts"

Add Build Step -> "Copy artifacts from another project"

Project name: MavenWeb\_Test

Build: Stable build only

Artifacts to copy: \*\*/\*

### Build Steps

The screenshot shows a configuration dialog for a 'Copy artifacts from another project' step. It includes fields for 'Project name' (MavenWeb\_Test), 'Which build' (Latest successful build), 'Build' (Stable build only checked), 'Artifacts to copy' (\*\*/\*), and 'Artifacts not to copy' (empty).

Copy artifacts from another project

Project name ?  
MavenWeb\_Test

! No such project 'MavenWeb\_Test'. Did you mean 'MavenJava\_Test'?

Which build ?  
Latest successful build

Stable build only

Artifacts to copy ?  
\*\*/\*

Artifacts not to copy ?

### Post-build Actions:

Add Post Build Action -> "Deploy WAR/EAR to a container"

WAR/EAR File: \*\*/\*.war

Context path: Webpath

Add container -> Tomcat 9.x remote

Credentials: Username: admin, Password: 1234

Tomcat URL: https://localhost:8085/

Apply and Save

**Deploy war/ear to a container**

WAR/EAR files ?

Context path ?

Containers

**Tomcat 9.x Remote**

Credentials

Tomcat URL ?

## Step 5: Create Pipeline View for MavenWeb

Click "+" beside "All" on the dashboard

Enter name: MavenWeb\_Pipeline

Select "Build pipeline view"

The screenshot shows the Jenkins dashboard with a search bar and user information at the top. Below it, a navigation bar shows 'Dashboard > All > New Item'. The main area is titled 'New Item' and asks for an item name, which is 'MavenWeb\_Pipeline'. It then prompts to select an item type. Three options are shown: 'Freestyle project' (classic job type), 'Maven project' (builds Maven projects), and 'Pipeline' (orchestrates long-running activities). The 'Pipeline' option is highlighted with a gray background.

## Pipeline Flow:

**Layout:** Based on upstream/downstream relationship

Initial job: MavenWeb\_Build

Apply and Save

## 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 ?

MavenWeb\_Build

### Trigger Options

Build Cards

Standard build card

Use the default build cards

Restrict triggers to most recent successful builds ?

OK

Apply

## Step 6: Run the Pipeline and Check Output

Click on the trigger “RUN” to run the pipeline

The screenshot shows the Jenkins Build Pipeline interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information. Below it, the main title is "Build Pipeline". A toolbar with icons for Run, History, Configure, Add Step, Delete, and Manage is visible. The pipeline consists of four stages:

- Pipeline #1**: Summary stage.
- #1 maven\_web\_build**: Triggered at 26-Nov-2024 7:54:24pm, took 12 sec, run by pavan.singireddy. Status is green.
- #1 maven\_web\_test**: Triggered at 26-Nov-2024 7:54:56pm, took 3.2 sec, run by pavan.singireddy. Status is green.
- #1 maven\_web\_deploy**: Triggered at 26-Nov-2024 7:55:06pm, took 1.5 sec, run by pavan.singireddy. Status is green.

The stages are connected by arrows indicating the flow from build to test to deployment.

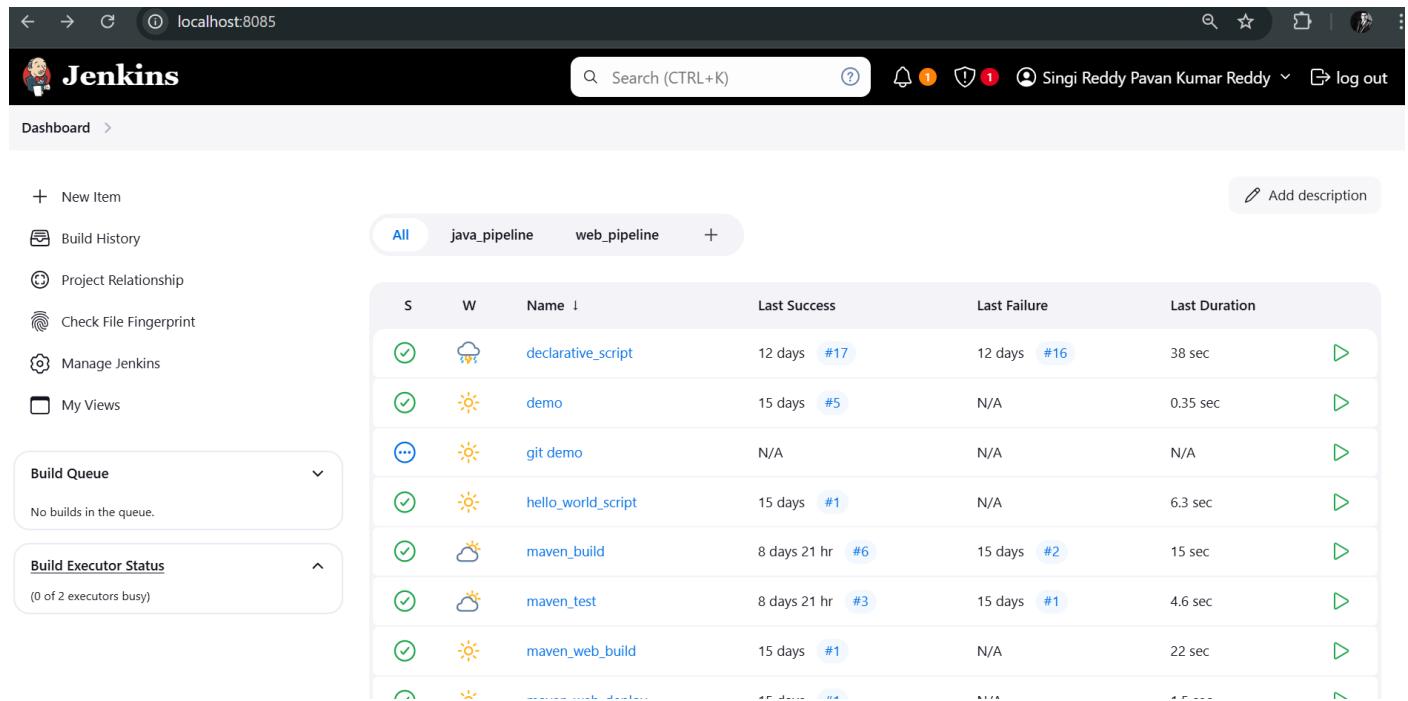
# Experiment – 6C

## Creation of Pipeline Script and Docker Installation

### Steps for Creation of Pipeline Script:

#### Step 1: Open Jenkins (localhost:8080)

Click on "New Item" (left side menu)



The screenshot shows the Jenkins dashboard at localhost:8080. On the left sidebar, there are links for 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. Below these are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0 of 2 executors busy). The main content area displays a table of pipeline projects. The columns are: S (Status), W (Work), Name (sorted by name), Last Success, Last Failure, and Last Duration. The projects listed are:

S	W	Name	Last Success	Last Failure	Last Duration
✓	Cloud	declarative_script	12 days #17	12 days #16	38 sec
✓	Sun	demo	15 days #5	N/A	0.35 sec
...	Sun	git demo	N/A	N/A	N/A
✓	Sun	hello_world_script	15 days #1	N/A	6.3 sec
✓	Cloud	maven_build	8 days 21 hr #6	15 days #2	15 sec
✓	Cloud	maven_test	8 days 21 hr #3	15 days #1	4.6 sec
✓	Sun	maven_web_build	15 days #1	N/A	22 sec
...	Sun	...	...	...	...

#### Step 2: Create Freestyle Project (e.g., Pipeline Script)

Enter project name (e.g., Pipeline Script)

Click "OK"

**Jenkins**Search (CTRL+K)?Bell iconShield iconUser icon: Singi Reddy Pavan Kumar ReddyLog out

Dashboard > All > New Item

## 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.

OK

## Step 3: Advance project option:

Choose pipeline script and copy the below code and paste it.

```
pipeline {
 agent any
 tools{
 maven 'MAVEN-HOME'
 }
 stages {
 stage('git repo & clean') {
 steps {
 //bat "rmdir /s /q mavenjava"
 bat "git clone provide your github link"
 bat "mvn clean -f mavenjava"
 }
 }
 stage('install') {
 steps {
 bat "mvn install -f mavenjava" #project name#
 }
 }
 }
}
```

```

stage('test') {
 steps {
 bat "mvn test -f mavenjava"
 }
}

stage('package') {
 steps {
 bat "mvn package -f mavenjava"
 }
}
}

```

The screenshot shows the Jenkins configuration interface for a pipeline job named 'declarative\_script'. The top navigation bar includes links for 'Dashboard', 'declarative\_script', and 'Configuration'. Below this, the 'Advanced Project Options' section is selected. The main area is titled 'Configure' and contains tabs for 'General', 'Advanced Project Options' (which is active), and 'Pipeline'. Under 'Pipeline', the 'Definition' dropdown is set to 'Pipeline script'. A large text area labeled 'Script' displays the Jenkinsfile code provided at the top of the page. At the bottom of this area are 'Save' and 'Apply' buttons.

```

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 maven_java"
10 bat "git clone https://github.com/pavansingireddy123/maven_java.git"
11 bat "mvn clean -f maven_java"
12 }
13 }
14 stage('install') {
15 steps {
16 bat "mvn install -f maven_java"
17 }
18 }
19 }
20 }

```

**Step 4:** click Save and Apply and Run the code

localhost:8085

# Jenkins

Search (CTRL+K)

Singi Reddy Pavan Kumar Reddy log out

Dashboard >

+ New Item

Build History

All

java\_pipeline

web\_pipeline

+

Add description

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

(0 of 2 executors busy)

S	W	Name	Last Success	Last Failure	Last Duration	
✓	cloud	declarative_script	12 days #17	12 days #16	38 sec	▶
✓	sun	demo	15 days #5	N/A	0.35 sec	▶
...	sun	git demo	N/A	N/A	N/A	▶
✓	sun	hello_world_script	15 days #1	N/A	6.3 sec	▶
✓	cloud	maven_build	8 days 21 hr #6	15 days #2	15 sec	▶
✓	cloud	maven_test	8 days 21 hr #3	15 days #1	4.6 sec	▶
✓	sun	maven_web_build	15 days #1	N/A	22 sec	▶
...	...	...	...	...	...	...

# Experiment – 7A

## Installation Of Docker for Windows

### Download Docker for Windows

## Installation Of Docker for Windows

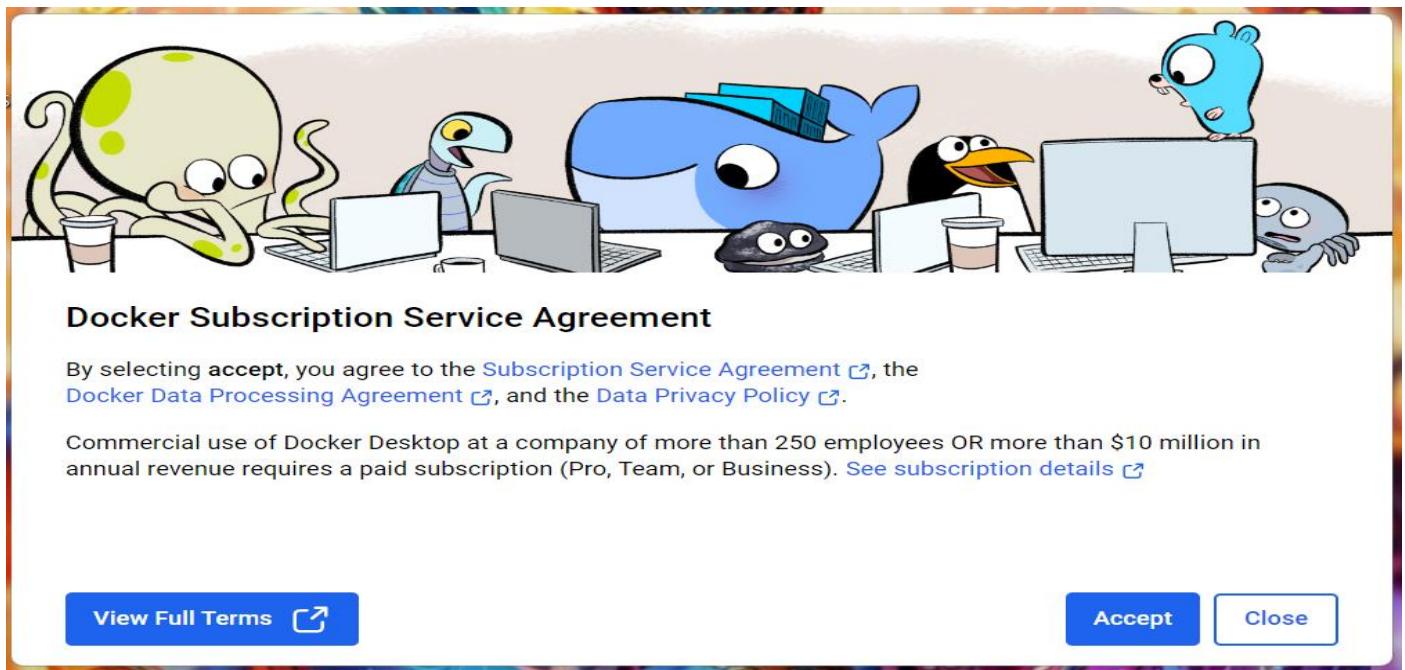
### Download Docker for Windows

The screenshot shows the Docker Docs website at [docs.docker.com/desktop/setup/install/windows-install/](https://docs.docker.com/desktop/setup/install/windows-install/). The left sidebar is titled 'Products' and includes sections for Docker Desktop, Setup, Install, Mac, Windows, Enterprise deployment, Linux, Run Docker Desktop for Windows..., Sign in, and Allowlist. The 'Windows' section is currently selected. The main content area has a breadcrumb navigation: Home / Manuals / Docker Desktop / Setup / Install / Windows. The title is 'Install Docker Desktop on Windows'. Below the title is a section titled 'Docker Desktop terms' which states: 'Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) requires a [paid subscription](#)'. There is also a note: 'This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.' Two download buttons are visible: 'Docker Desktop for Windows - x86\_64' and 'Docker Desktop for Windows - Arm (Beta)'. On the right side, there are links for 'Edit this page', 'Request changes', 'Table of contents', 'System requirements', and 'Where to go next'. A 'Give feedback' button is located at the bottom right.

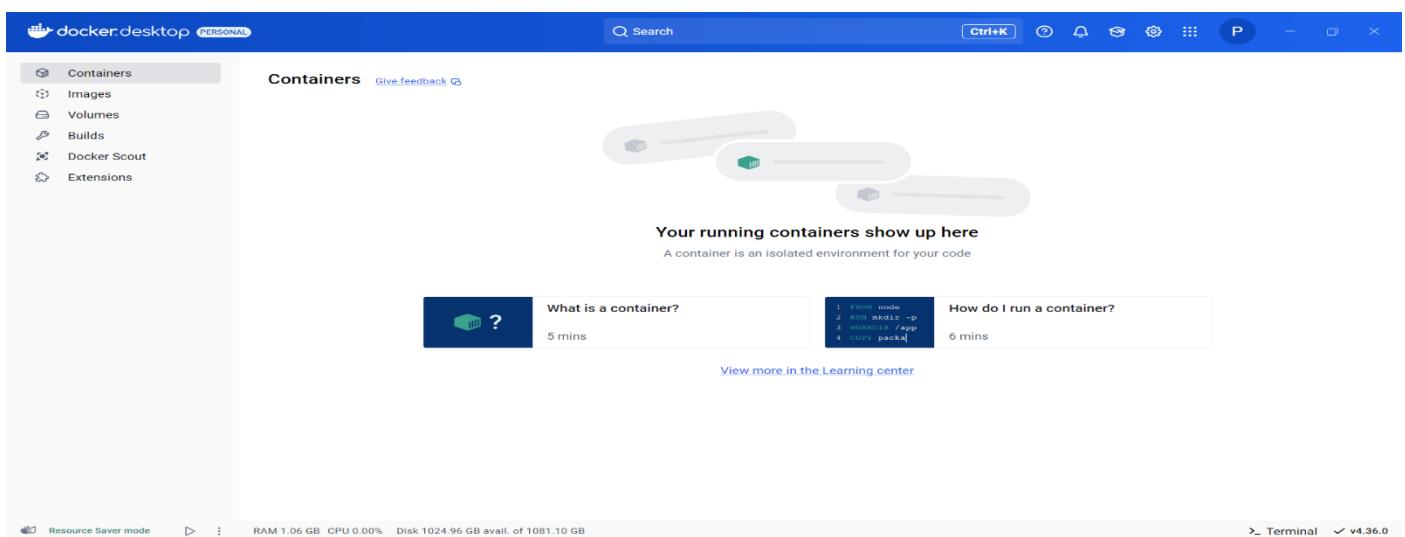
### Download Docker for Windows

The screenshot shows the Docker Hub website at [hub.docker.com/](https://hub.docker.com/). The top navigation bar includes 'Explore', 'Repositories' (which is the active tab), 'Organizations', 'Usage', and a search bar. A 'Recent download history' sidebar shows a file named 'Docker Desktop Installer.exe' (5.9/495 MB, 14 minutes left). The main content area features a blue banner with the text 'Welcome to Docker' and 'Download the desktop application'. It includes a 'Download for Windows' button and notes that it's also available for Mac and Linux. Below the banner are three cards: 'Create a Repository' (Push container images to a repository on Docker Hub.), 'Docker Hub Basics' (Watch the guide on how to create and push your first image into a Docker Hub repository.), and 'Language-Specific Guides' (Learn how to containerize language-specific applications using Docker.). At the bottom, a cookie consent banner asks for permission to store cookies, with options for 'Cookies Settings', 'Reject All', and 'Accept All Cookies'.

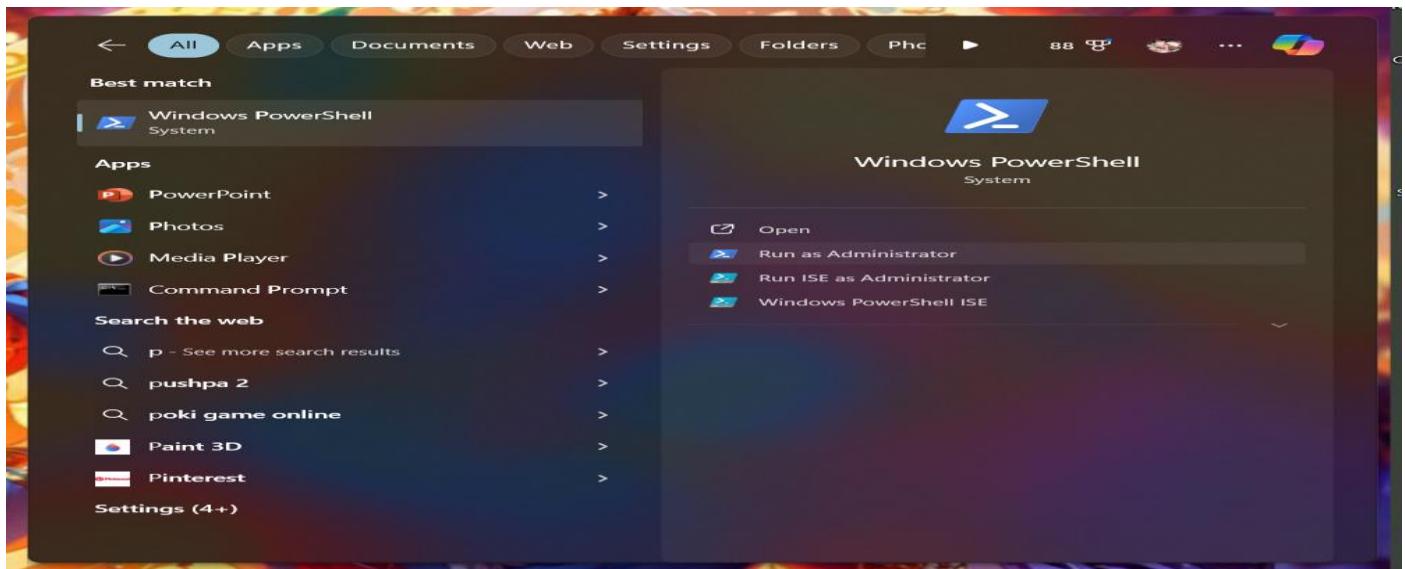
Accept the license → click Accept



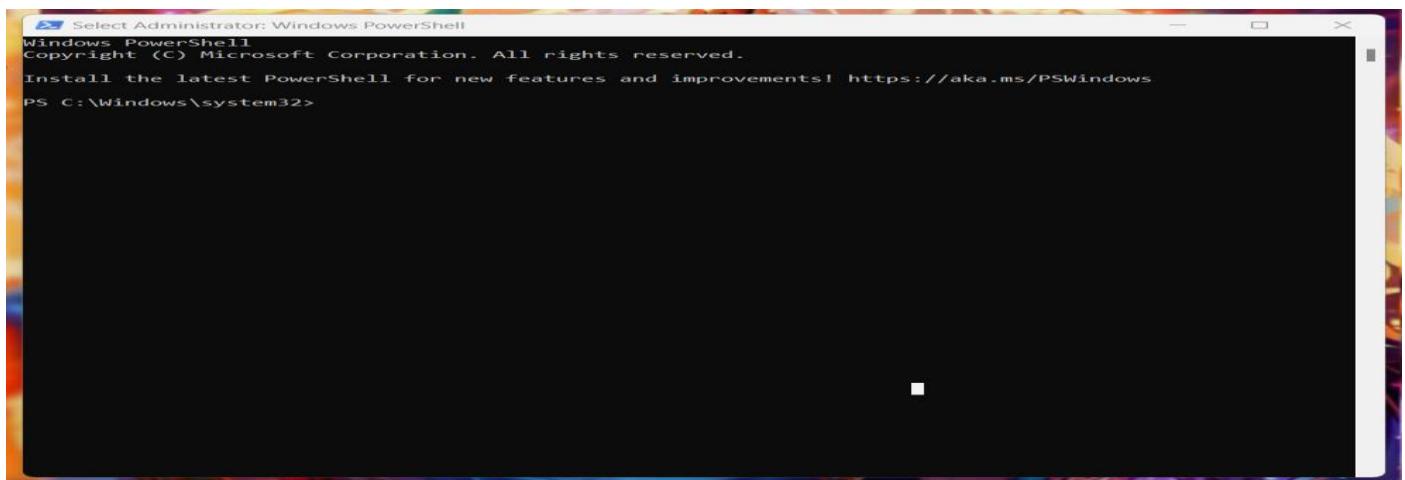
## The local Docker Hub is created



## Open Windows PowerShell as shown from Start



Type the following command, to set WSL as default version



Click on hub.docker.com and signup as shown

Sign In into Docker Hub



## Create your account

Email

We suggest signing up with your work email address.

Username

Password



Send me occasional product updates and announcements.

OR



Continue with Google

## Understanding Docker and Redis

### What is Docker?

Docker is a tool that makes running applications easy by packaging everything (code, libraries,

tools) into containers.

- Containers are like lightweight virtual machines but more efficient because they share the host's system resources.

What is Redis?

Redis (Remote Dictionary Server) is:

- A super-fast database that stores data in memory (not on a disk).
- Commonly used for:
  - Caching: Storing temporary data for quick access.
  - Real-time applications: Like live chat, analytics, or leaderboards.
  - Data structures: Redis supports lists, hashes, sets, and more.

Example:

- Save data: Use the key "name" and the value "Alice".
- Retrieve data: Ask Redis for "name", and it will give you "Alice" instantly.

## Setting Up Docker

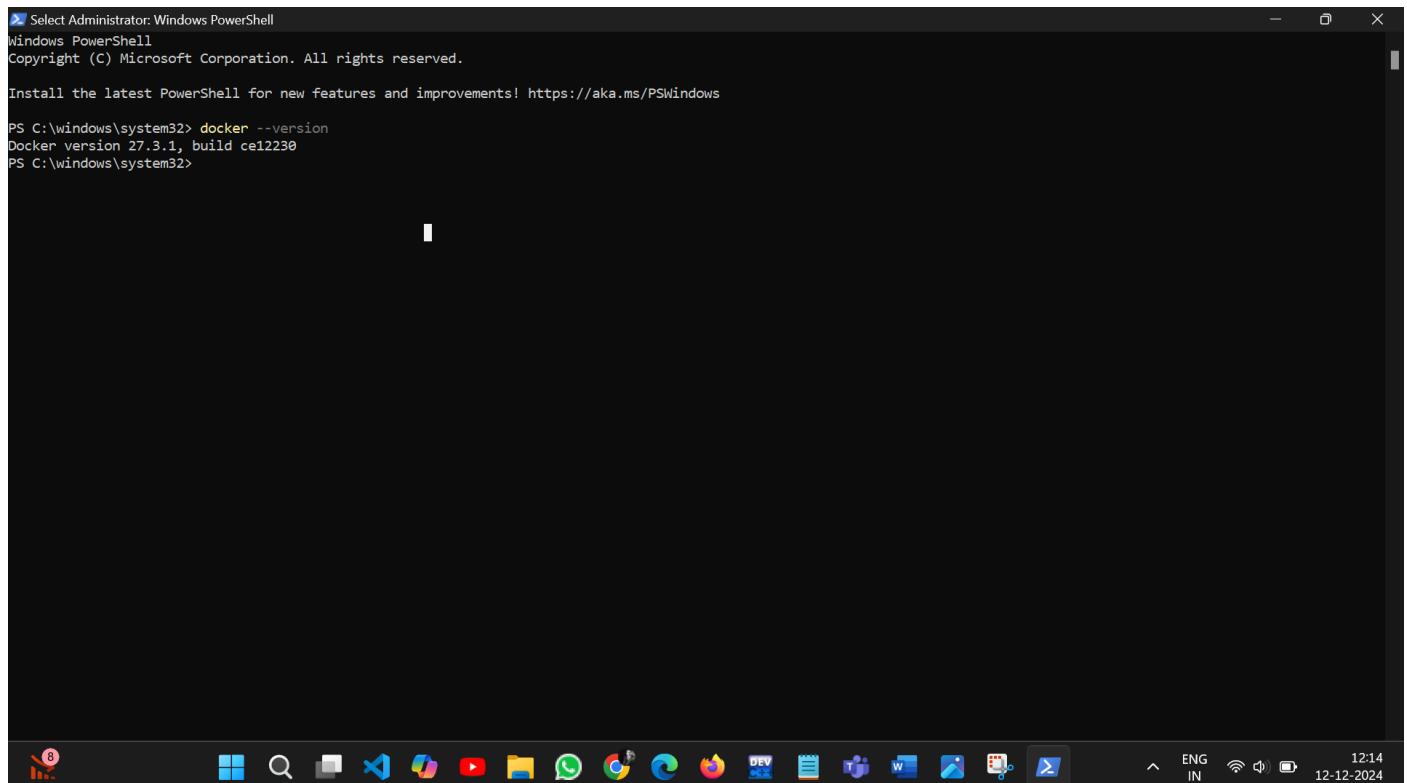
### Step 1: Choose the Right Terminal

- Windows: Use Git Bash or PowerShell (Git Bash is preferred for Docker commands).
- Mac/Linux/Ubuntu: Use the built-in Terminal.

### Step 2: Verify Docker Installation

Run this command to check if Docker is installed:

**docker --version**



```
PS Select 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>
```

# Experiment – 7B

## Docker CLI Commands with hello-world

Why Use hello-world?

The hello-world image is a basic test to ensure Docker is working correctly.

Step 1: Pull the hello-world Image

**Command:**

**docker pull hello-world**

```
PS C:\windows\system32> docker version
Client:
 Version: 27.3.1
 API version: 1.47
 Go version: go1.22.7
 Git commit: ce12230
 Built: Fri Sep 20 11:42:27 2024
 OS/Arch: windows/amd64
 Context: desktop-linux

Server: Docker Desktop 4.36.0 (175267)
Engine:
 Version: 27.3.1
 API version: 1.47 (minimum version 1.24)
 Go version: go1.22.7
 Git commit: 41ca978
 Built: Fri Sep 20 11:41:11 2024
 OS/Arch: linux/amd64
 Experimental: false
containerd:
 Version: 1.7.21
 GitCommit: 472731909fa34bd7bc9c087e4c27943f9835f111
runc:
 Version: 1.1.13
 GitCommit: v1.1.13-0-g58aa920
docker-init:
 Version: 0.19.0
 GitCommit: de40ad0
PS C:\windows\system32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
```

The screenshot shows a Windows PowerShell window with the title "Select Administrator: Windows PowerShell". It displays the output of two Docker commands: "docker version" and "docker images". The "docker version" command shows client and server details, including Docker Desktop version 4.36.0 and Docker Engine version 27.3.1. The "docker images" command lists the available Docker images. The taskbar at the bottom shows various pinned icons like File Explorer, Task View, and Edge browser.

What It Does:

- Downloads the hello-world image from Docker Hub (Docker's app store).

Where to Run:

- Open your terminal (Git Bash for Windows or Terminal for Mac/Linux).
- Run the command from any folder.

Step 2: Run the hello-world Image

**Command:**

**docker run hello-world**

```
PS Select Administrator: Windows PowerShell
exit
PS C:\windows\system32> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Download complete
Digest: sha256:5b3cc85e16e3058003c13b7821318369dad01dac3dbb877aac3c28182255c724
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
PS C:\windows\system32>
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/u/your-docker-id
```



6 21°C Haze

10:44  
ENG IN 12-12-2024

## What It Does:

- Creates and runs a container from the hello-world image.
- Displays a message to confirm that Docker is installed and working.

## Output Example:

Hello from Docker!

This message shows that your installation appears to be working correctly.

## Step 3: View All Containers

### Command:

**docker ps -a**

```
PS Select Administrator: Windows PowerShell
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 (amd64)
3. The Docker daemon created a new container from that image which runs the
 executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
 to your terminal.

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

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

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

PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 12 minutes ago Exited (0) 5 minutes ago
64825962355b ubuntu "/bin/bash" 29 minutes ago Exited (130) 6 minutes ago
PS C:\windows\system32> docker rm 64825962355b
64825962355b
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 13 minutes ago Exited (0) 13 minutes ago
PS C:\windows\system32>


ENG IN 10:47
12-12-2024
```

## What It Does:

- Lists all containers (running and stopped).
- The hello-world container will show as "Exited" because it stops after displaying the message.

## Step 4: Remove the hello-world Container

### Command:

**docker rm [container-id]**

```
PS Select Administrator: Windows PowerShell
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 (amd64)
3. The Docker daemon created a new container from that image which runs the
 executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
 to your terminal.

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

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

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

PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 12 minutes ago Exited (0) 5 minutes ago
64825962355b ubuntu "/bin/bash" 29 minutes ago Exited (130) 6 minutes ago
PS C:\windows\system32> docker rm 64825962355b
64825962355b
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 13 minutes ago Exited (0) 13 minutes ago
PS C:\windows\system32>


ENG IN 10:47
12-12-2024
```

## What It Does:

- Deletes the container to free up space.
- Replace [container-id] with the actual ID from docker ps -a.

## Step 5: Remove the hello-world Image

### Command:

```
docker rmi hello-world
```

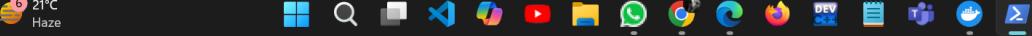
```
PS Select Administrator: Windows PowerShell
exit
PS C:\windows\system32> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Download complete
Digest: sha256:5b3cc85e16e3058003c13b7821318369dad01dac3dbb877aac3c28182255c724
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
PS C:\windows\system32>
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:
Get Docker ID
```



The taskbar icons include: Weather (21°C), Haze, Start button, Search, Task View, File Explorer, Microsoft Edge, WhatsApp, Google Chrome, Mozilla Firefox, DEV Community, Microsoft Word, Microsoft Teams, Microsoft Edge Dev, and Microsoft Edge Canary.

System tray icons: ENG IN, Wi-Fi, Battery (12-12-2024, 10:44).

## What It Does:

- Deletes the hello-world image if you no longer need it

# Experiment – 7C

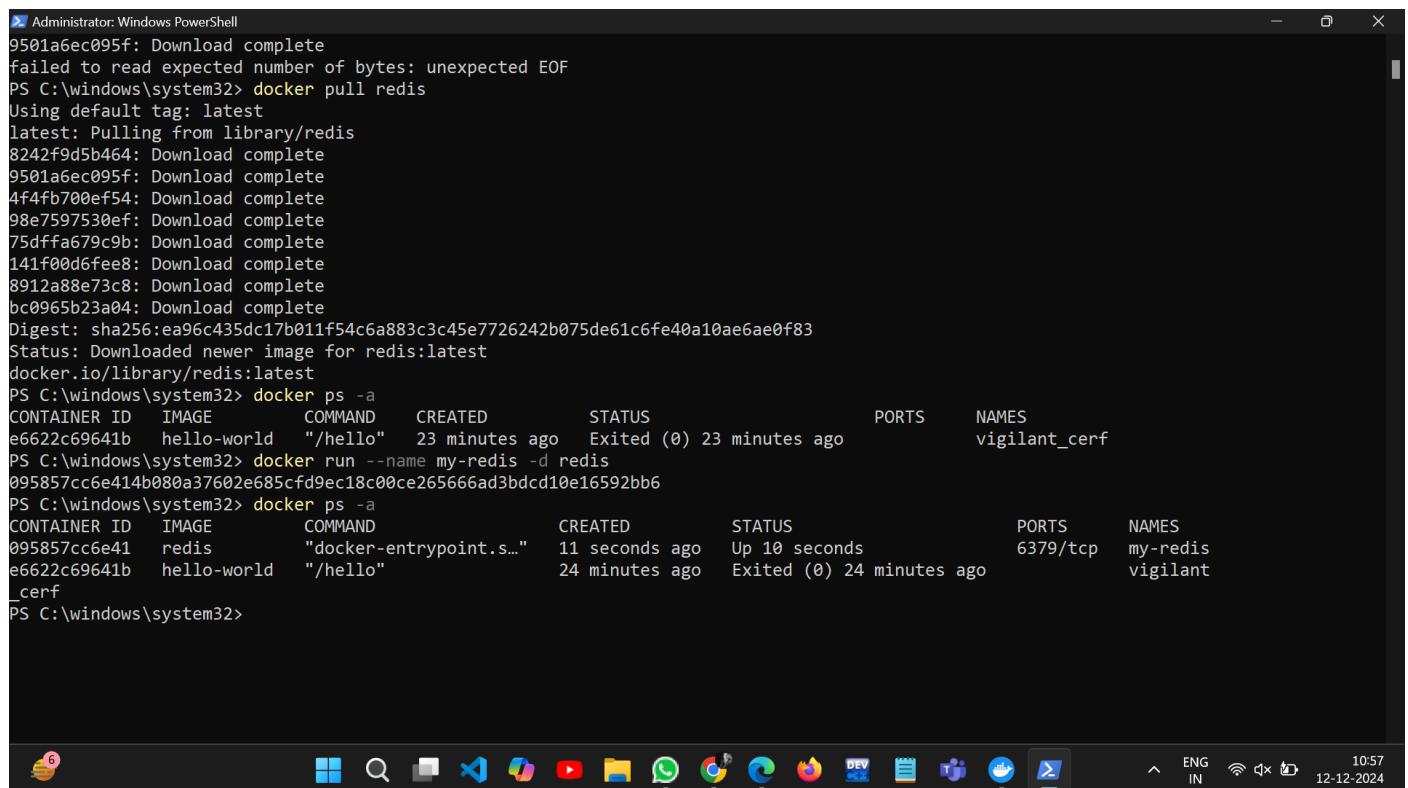
## Why Use redis?

Redis is a powerful, real-world example of a service often run using Docker.

### Step 1: Pull the redis Image

#### Command:

```
docker pull redis
```



```
Administrator: Windows PowerShell
9501a6ec095f: Download complete
failed to read expected number of bytes: unexpected EOF
PS C:\windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8242f9d5b464: Download complete
9501a6ec095f: Download complete
4f4fb700ef54: Download complete
98e7597530ef: Download complete
75dffaa679c9b: Download complete
141f00d6fee8: Download complete
8912a88e73c8: Download complete
bc0965b23a04: Download complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago
PS C:\windows\system32> docker run --name my-redis -d redis
095857cc6e414b080a37602e685cf9ec18c00ce265666ad3bcd10e16592bb6
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
095857cc6e41 redis "docker-entrypoint.s..." 11 seconds ago Up 10 seconds 6379/tcp my-redis
e6622c69641b hello-world "/hello" 24 minutes ago Exited (0) 24 minutes ago
vigilant_cerf
PS C:\windows\system32>
```

## What It Does:

- Downloads the official redis image from Docker Hub to your system.

### Step 2: Run a Redis Container

#### Command:

```
docker run --name my-redis -d redis
```

```
Administrator: Windows PowerShell
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago
PS C:\windows\system32> docker run --name my-redis -d redis
095857cc6e414b080a37602e685cf9ec18c00ce265666ad3bcd10e16592bb6
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
095857cc6e41 redis "docker-entrypoint.s..." 11 seconds ago Up 10 seconds 6379/tcp my-redis
e6622c69641b hello-world "/hello" 24 minutes ago Exited (0) 24 minutes ago
_cerf
PS C:\windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
095857cc6e41 redis "docker-entrypoint.s..." 45 seconds ago Up 44 seconds 6379/tcp my-redis
PS C:\windows\system32> docker exec -it my-redis redis-cli
127.0.0.1:6379> SET name "Pavan"
OK
127.0.0.1:6379> GET name
"Pavan"
127.0.0.1:6379> docker stop my-redis
(error) ERR unknown command 'docker', with args beginning with: 'stop' 'my-redis'
127.0.0.1:6379> docker stop my-redis
(error) ERR unknown command 'docker', with args beginning with: 'stop' 'my-redis'
127.0.0.1:6379> docker ps
(error) ERR unknown command 'docker', with args beginning with: 'ps'
127.0.0.1:6379> exit
PS C:\windows\system32> docker stop my-redis
my-redis
PS C:\windows\system32>
```

## What It Does:

- Creates and starts a container named my-redis from the redis image.
- The -d flag runs the container in the background.

## Step 3: Check Running Containers

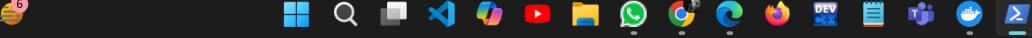
### Command:

**docker ps**

```

Administrator: Windows PowerShell
9501a6ec095f: Download complete
failed to read expected number of bytes: unexpected EOF
PS C:\windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8242f9d5b464: Download complete
9501a6ec095f: Download complete
4f4fb700ef54: Download complete
98e7597530ef: Download complete
75dfffa679c9b: Download complete
141f00d6fee8: Download complete
8912a88e73c8: Download complete
bc0965b23a04: Download complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago
PS C:\windows\system32> docker run --name my-redis -d redis
095857cc6e41b080a37602e685cf9e18c00ce265666ad3bcd10e16592bb6
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
095857cc6e41b redis "docker-entrypoint.s..." 11 seconds ago Up 10 seconds 6379/tcp my-redis
e6622c69641b hello-world "/hello" 24 minutes ago Exited (0) 24 minutes ago
_cerf
PS C:\windows\system32>

```



10:57  
12-12-2024

## What It Does:

- Lists all running containers.
- You should see the Redis container (my-redis) in the list.

## Step 4: Access Redis

### Command:

```
docker exec -it my-redis redis-cli
```

or

```
winpty docker exec -it myredis redis-cli
```

## What It Does:

- Opens the Redis command-line tool (redis-cli) inside the container.
- You can now send commands directly to the Redis server.
- winpty: This command makes Git Bash handle the terminal interaction correctly, allowing you to run commands that require user input.
- docker exec -it myredis redis-cli: This runs the Redis command-line interface (redis cli) inside the running myredis container.

### Example Redis Commands:

```
127.0.0.1:6379> SET name "Alice"
```

**OK**

**127.0.0.1:6379> GET name**

**"Alice"**

```
Administrator: Windows PowerShell
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e6622c69641b hello-world "/hello" 23 minutes ago Exited (0) 23 minutes ago
PS C:\windows\system32> docker run --name my-redis -d redis
095857cc6e41b080a37602e685cf9ec18c00ce265666ad3bcd10e16592bb6
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
095857cc6e41b redis "docker-entrypoint.s..." 11 seconds ago Up 10 seconds 6379/tcp my-redis
e6622c69641b hello-world "/hello" 24 minutes ago Exited (0) 24 minutes ago
_cerf
PS C:\windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
095857cc6e41b redis "docker-entrypoint.s..." 45 seconds ago Up 44 seconds 6379/tcp my-redis
PS C:\windows\system32> docker exec -it my-redis redis-cli
127.0.0.1:6379> SET name "Pavan"
OK
127.0.0.1:6379> GET name
"Pavan"
127.0.0.1:6379> docker stop my-redis
(error) ERR unknown command 'docker', with args beginning with: 'stop' 'my-redis'
127.0.0.1:6379> docker stop my-redis
(error) ERR unknown command 'docker', with args beginning with: 'stop' 'my-redis'
127.0.0.1:6379> docker ps
(error) ERR unknown command 'docker', with args beginning with: 'ps'
127.0.0.1:6379> exit
PS C:\windows\system32> docker stop my-redis
my-redis
PS C:\windows\system32>
```

## Step 5: Stop the Redis Container

**Command:**

**docker stop my-redis**

**What It Does:**

- Stops the Redis container but doesn't delete it.

## Step 6: Restart the Redis Container

**Command:**

**docker start my-redis**

**What It Does:**

- Restarts the stopped container.

**• Command:**

**docker stop my-redis**

# Experiment – 7D

## Docker Compose

### 1. Writing a Basic docker-compose.yml File

#### Step 1: Create a Folder

Go to your desktop or any folder you prefer.

Right-click and select New Folder.

Name the folder my\_docker\_project.

#### Step 2: Open a Text Editor

Open Visual Studio Code, Notepad, or any other text editor.

Create a new file.

#### Step 3: Write the YAML Configuration

Here's a simple example of a docker-compose.yml file for WordPress and MySQL:

```
version: '3.1'

services:
 db:
 image: mysql:5.7
 container_name: mysql_container
 environment:
 MYSQL_ROOT_PASSWORD: rootpassword
 MYSQL_DATABASE: wordpress_db
 MYSQL_USER: wordpress_user
 MYSQL_PASSWORD: wordpress_pass
 volumes:
 - db_data:/var/lib/mysql

 wordpress:
 depends_on:
 - db
 image: wordpress:latest
 container_name: wordpress_container
 ports:
 - "8000:80"
 environment:
```

**WORDPRESS\_DB\_HOST: db:3306**

**WORDPRESS\_DB\_USER: wordpress\_user**

**WORDPRESS\_DB\_PASSWORD: wordpress\_pass**

**WORDPRESS\_DB\_NAME: wordpress\_db**

**volumes:**

- ./wordpress\_data:/var/www/html

**volumes:**

### **3 RKR21 SOFTWARE ENGINEERING LAB CSE/IT/CSM/CSD III/I**

**db\_data:**

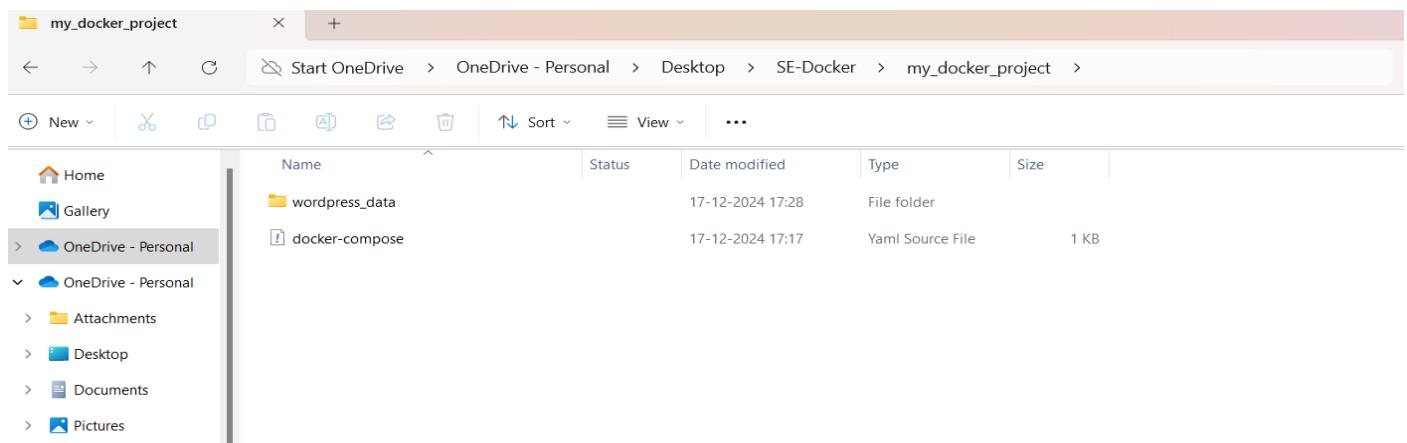
```
File Edit View

version: '3.1'
services:
 db:
 image: mysql:5.7
 container_name: mysql_container
 environment:
 MYSQL_ROOT_PASSWORD: rootpassword
 MYSQL_DATABASE: wordpress_db
 MYSQL_USER: wordpress_user
 MYSQL_PASSWORD: wordpress_pass
 volumes:
 - db_data:/var/lib/mysql
 wordpress:
 depends_on:
 - db
 image: wordpress:latest
 container_name: wordpress_container |
 ports:
 - "8000:80"
 environment:
 WORDPRESS_DB_HOST: db:3306
 WORDPRESS_DB_USER: wordpress_user
 WORDPRESS_DB_PASSWORD: wordpress_pass
 WORDPRESS_DB_NAME: wordpress_db
 volumes:
 - ./wordpress_data:/var/www/html
 volumes:
 db_data:
```

## **2. Saving the File**

Save the file as **docker-compose.yml**.

Place it in the **my\_docker\_project** folder.



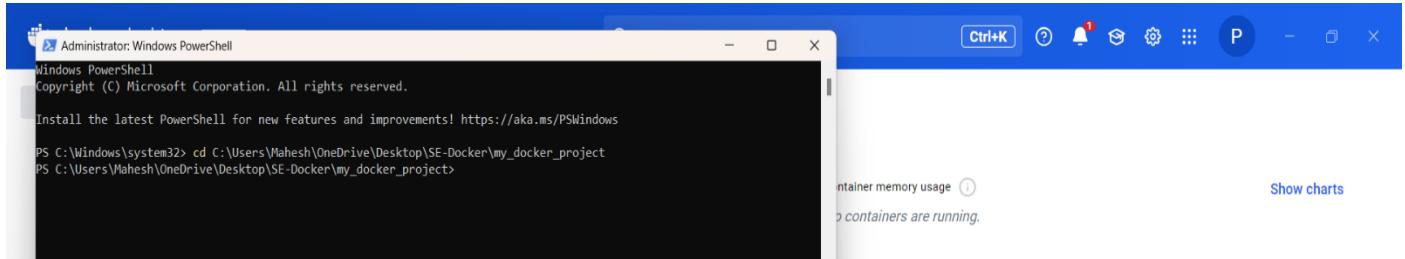
### 3. Running the Setup

#### Step 1: Open Command Line

Open PowerShell or Command Prompt.

Navigate to the my\_docker\_project folder:

```
cd path_to_my_docker_project
```



```
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

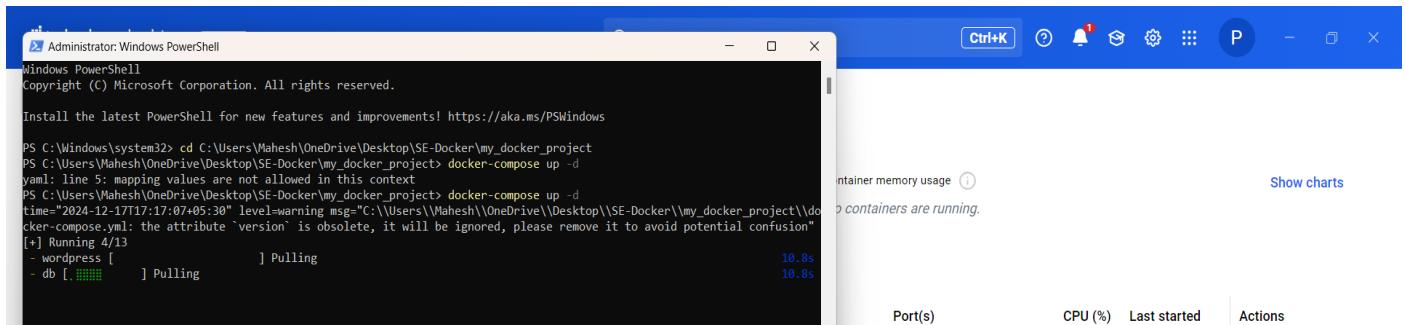
PS C:\Windows\system32> cd C:/Users/Mahesh/OneDrive/Desktop/SE-Docker/my_docker_project
PS C:/Users/Mahesh/OneDrive/Desktop/SE-Docker/my_docker_project>
```

The PowerShell window shows the command being run. To the right, a Docker container monitoring interface is visible, showing 'Container memory usage' and '0 containers are running.'

#### Step 2: Start the Containers

Run:

```
docker-compose up -d
```



```
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/Mahesh/OneDrive/Desktop/SE-Docker/my_docker_project
PS C:/Users/Mahesh/OneDrive/Desktop/SE-Docker/my_docker_project> docker-compose up -d
yaml: line 5: mapping values are not allowed in this context
PS C:/Users/Mahesh/OneDrive/Desktop/SE-Docker/my_docker_project> docker-compose up -d
time="2024-12-17T17:17:07+05:30" level=warning msg="C:/Users/Mahesh/OneDrive/Desktop/SE-Docker/my_docker_project/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/13
- wordpress [] Pulling
- db [██████] Pulling
10.8s
10.8s
```

The PowerShell window shows the command being run. To the right, a Docker container monitoring interface is visible, showing 'Container memory usage' and '0 containers are running.' Below the monitor are tabs for 'Port(s)', 'CPU (%)', 'Last started', and 'Actions'.

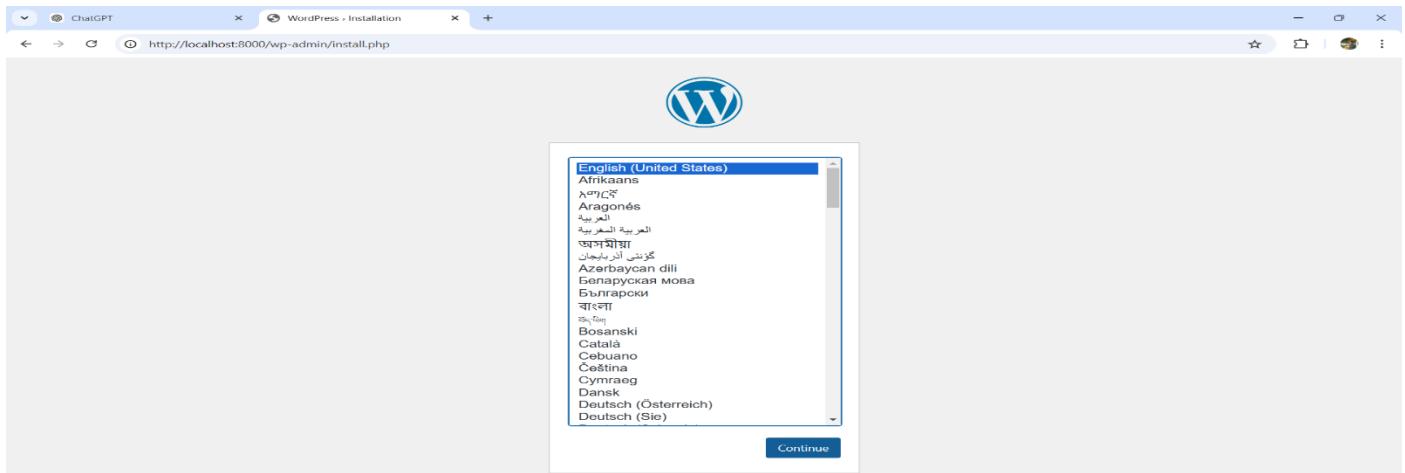
This command reads the docker-compose.yml file and creates both the WordPress and MySQL containers.

-d runs the containers in the background.

### 4. Accessing the Application

Open your web browser.

Go to <http://localhost:8000>



Follow the WordPress setup wizard to complete the installation:

Site Name.

## Admin Username and Password.

### Email Address.

The screenshot shows the WordPress installation welcome screen. It displays the following fields:

- Site Title:** WordPress
- Username:** Penkula Mahesh Kumar
- Password:** Maheshkumar@1213# (Strength: Strong)
- Your Email:** penkulamaheshkumar4@gmail.com
- Search engine visibility:** Discourage search engines from indexing this site (checkbox checked)

At the bottom right is a blue "Install WordPress" button.

## 8. Managing Containers

### Stop the Containers

To stop the containers without removing them:

**docker-compose stop**

### Start Again

To restart the containers:

**docker-compose start**

### Remove Containers

To stop and remove everything:

**docker-compose down**

The screenshot shows a Windows PowerShell window with administrator privileges running the command `docker-compose stop`. The output indicates that the MySQL container has stopped. The user then runs `docker-compose start`, which starts both the MySQL and WordPress containers. Finally, the user runs `docker-compose down`, which stops and removes both containers.

Below the PowerShell window, a screenshot of a WordPress dashboard is visible, showing the "Howdy, Penkula Mahesh Kumar" greeting and some site management options.

```
[+] Running 4/4
 Network my_docker_project_default Created
 Volume "my_docker_project_db_data" Created
 Container mysql_container Started
 Container wordpress_container Started
PS C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project> docker-compose stop
time="2024-12-17T17:27:51+05:30" level=warning msg="C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Stopping 2/2
 Container wordpress_container Stopped
 Container mysql_container Stopped
PS C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project> docker-compose start
time="2024-12-17T17:28:12+05:30" level=warning msg="C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 2/2
 Container mysql_container Started
 Container wordpress_container Started
PS C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project> docker-compose down
time="2024-12-17T17:28:32+05:30" level=warning msg="C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Stopping 2/2
 Container wordpress_container Removed
 Container mysql_container Removed
 Network my_docker_project_default Removed
PS C:\Users\Mahesh\OneDrive\Desktop\SE-Docker\my_docker_project>
```

http://localhost:8000/wp-admin/

Howdy, Penkula Mahesh Kumar

Dashboard

Welcome to WordPress!

Learn more about the 6.7.1 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

Site Health Status

No information yet...

Quick Draft

Title

# Experiment – 7E

## Minikube and Creation of virtual machine for Ubuntu OS and Deploying the Web Application

Steps to install the minikube

Search for minikube install in browser

minikube installation

All Images Videos News

About 106,000 search results

minikube.sigs.k8s.io - docs - start

minikube start

Jul 11, 2024 · Learn how to install and use minikube, a tool that lets you run Kubernetes on your local machine. Follow the steps to download, start, and interact with your cluster, and deploy...

Related searches

- minikube installation on windows 10
- minikube install windows 10 download
- minikube installation on windows 11
- reinstall minikube
- installing kubernetes with minikube

View all

Check for the pre-requisites

ubectl  
accessing apps  
ddns  
configuration  
dashboard  
pushing images  
proxies and PNs  
registries  
certificates

### What you'll need

- 2 CPUs or more
- 2GB of free memory
- 20GB of free disk space
- Internet connection
- Container or virtual machine manager, such as: Docker, QEMU, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox, or VMware Fusion/Workstation

Select the operating system here I am going with windows

minikube

Community GitHub Search this site...

Documentation Get Started! Handbook Addons Configuration

## 1 Installation

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: Windows

Architecture: x86-64

Release type: Stable

Installer type: .exe download

To install the latest minikube stable release on x86-64 Windows using .exe download:

- Download and run the installer for the latest release.  
Or if using PowerShell, use this command:  

```
New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/x86_64-windows-minikube.exe'
```
- Add the minikube.exe binary to your PATH.  
Make sure to run PowerShell as Administrator.

Already we have installed the docker than proceed for next steps

Copy the commands in powershell

 minikube

Community GitHub  Search this site...

Search this site... 

Documentation 

Get Started! 

Handbook 

- Basic controls
- Deploying apps
- Kubectl
- Accessing apps

Addons 

- Configuration
- Dashboard
- Pushing images
- Proxies and VPNs
- Registries
- Certificates
- Offline usage
- Host access
- Network Policy
- Persistent Volumes
- Mounting filesystems

To install the latest minikube **stable** release on **x86\_64 Windows** using **.exe download**:

1. Download and run the installer for the latest release.  
Or if using `PowerShell`, use this command:  

```
Type Directory .Force
$kube.exe -Uri "https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe" -UseBasicParsing
4
```


2. Add the `minikube.exe` binary to your `PATH`.  
*Make sure to run PowerShell as Administrator.*  

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

If you used a terminal (like powershell) for the installation, please close the terminal and reopen it before running minikube.

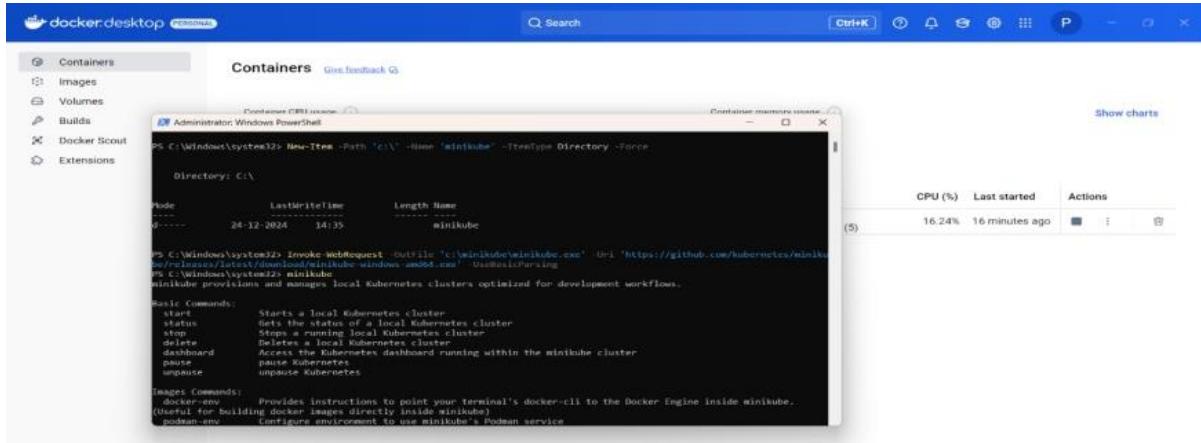
## 2 Start your cluster

From a terminal with administrator access (but not logged in as root), run:

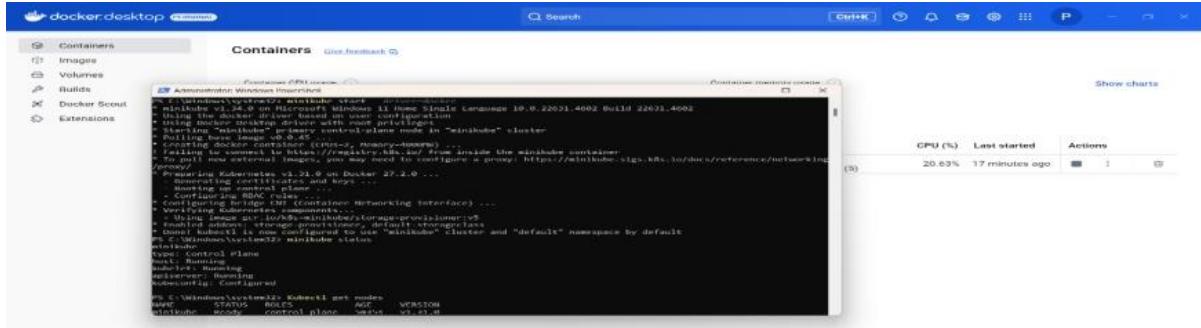
```
minikube start
```

  
What you'll need  
Take the next step

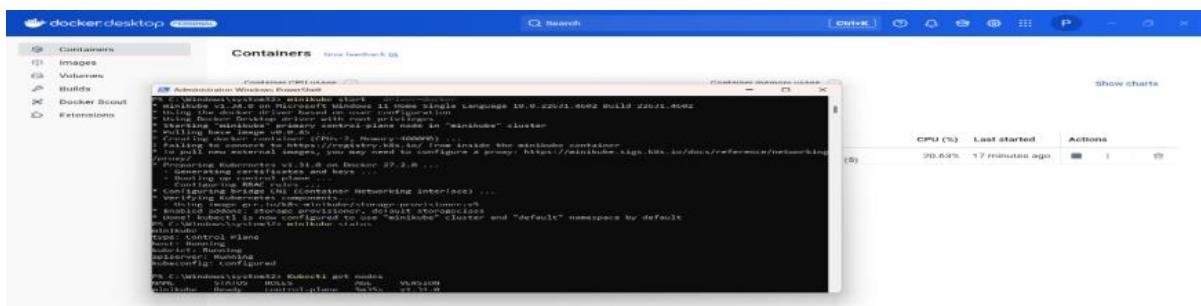
Once done with commands start the minikube



## Minikube start



## Minikube status



## Kubectl get nodes

The screenshot shows the Docker Desktop application window. The left sidebar has tabs for 'Containers', 'Images', 'Volumes', 'Builds', 'Docker Scouting', and 'Extensions'. The 'Containers' tab is selected. In the center, there's a terminal window titled 'Administrator: Windows PowerShell' with the command 'kubectl get pods' run, showing one pod named 'nginx-deployment-7f7f7' in the 'READY' state. To the right of the terminal is a chart titled 'nginx-deployment' showing CPU usage over time. Below the terminal is a table with columns 'NAME', 'READY', 'UP-TO-DATE', 'AVAILABLE', and 'AGE'. It lists the single pod from the terminal. At the bottom, there's a detailed view of the pod with fields like 'Name', 'Namespace', 'Priority', 'Service Account', 'Labels', 'Annotations', and 'Status'.

Kubectl get pods

## Experiment – 7F

## **Creating the nginx deployment**

## **1. To create the deployment for nginx server**

```
Kubectl create deployment <deployment name> --image=nginx
```

A screenshot of the Docker Desktop interface. The left sidebar shows 'Containers' selected. A central terminal window titled 'Administrator: Windows PowerShell' is open, displaying the command 'minikube start'. The terminal output shows the creation of a deployment named 'hello-minikube' with one pod running. The right side of the screen shows the 'hello-minikube' container details, including its CPU usage (26.52%), last started time (26 minutes ago), and actions menu.

## **2. To check whether deployment is created or not**

## Kubectl get deployment

```
[Administrator: Windows PowerShell] PS C:\Windows\system32\WindowsPowerShell\v1.0> az deployment create --resource-group "my-resource-group" --name "my-deployment" --template-file "my-deployment.json" --parameters file=~/my-deployment.json
{
 "id": "my-deployment",
 "status": "Running"
}
```

### 3. To check the pod status

Kubectl get pod

The screenshot shows the Docker Desktop application window. The 'Containers' tab is selected, displaying a list of running Docker containers. Each container entry includes the name, image, status, restart count, and CPU usage. A 'Show charts' button is located in the top right corner of the container list area.

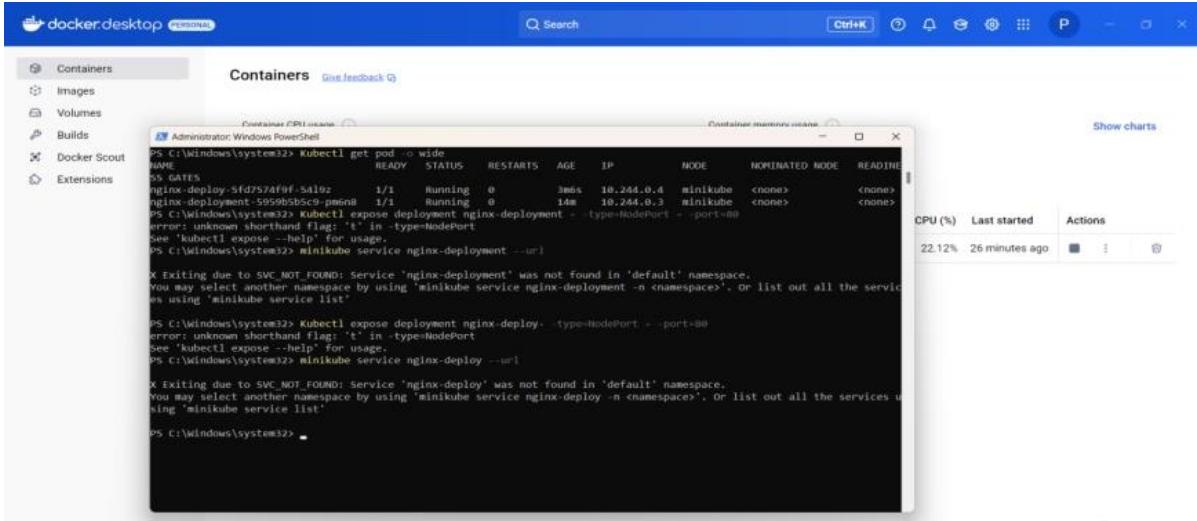
#### **4. To check service**

## Kubectl get service

```
C:\Windows\system32> kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-ingress-5959bbcc9c-pm6n3 1/1 Running 0 10m
nginx-ingress-5959bbcc9c-wt49z 1/1 Running 0 10m
C:\Windows\system32> kubectl create deployment nginx-deployment --image=nginx-ingress
Deployment.apps/nginx-deployment created
C:\Windows\system32> kubectl get pods
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-deployment 1/1 1 1 23s
C:\Windows\system32> kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-deployment-5d7574f9f-54t9z 1/1 Running 0 46s
nginx-deployment-5d7574f9f-pm6n3 1/1 Running 0 46s
C:\Windows\system32> minikube get service
NAME PORT(S) SELECTOR IP(s)
nginx-ingress 80:32461/TCP app=nginx-ingress 192.168.49.2
Run `minikube help` for usage.
C:\Windows\system32> kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) SELECTOR
Namespace: default
Priority: 0
minikube ClusterIP 192.168.49.2
Nodes: minikube[192.168.49.2]
Start Time: wed, 28 dec 2022 20:25:53 +05:00
Labels: app=nginx-ingress
Annotations: none
Status: running
IP: 10.244.0.4
```

## 5. To get pod information in more detail

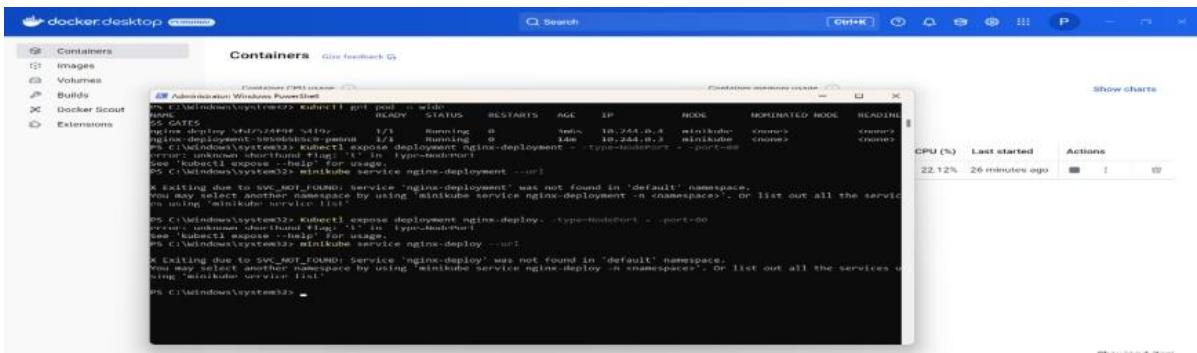
Kubectl get pod -o wide



```
PS C:\Windows\system32> kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS
SS GATES
nginx-deployment-55d757af9f-54l0z 1/1 Running 0 3m6s 10.244.0.4 minikube <none> <none>
nginx-deployment-59595b5c9-pm6n8 1/1 Running 0 1m8s 10.244.0.3 minikube <none> <none>
PS C:\Windows\system32> kubectl expose deployment nginx-deployment - --type=NodePort - --port=80
error: unknown shorthand flag: 't' in --type=NodePort
See 'kubectl expose --help' for usage.
PS C:\Windows\system32> minikube service nginx-deployment --url
X Exiting due to SVC_NOT_FOUND: Service 'nginx-deployment' was not found in 'default' namespace.
You may select another namespace by using 'minikube service nginx-deployment -n <namespace>', or list out all the services using 'minikube service list'
PS C:\Windows\system32>
```

## 6. To access the nginx page

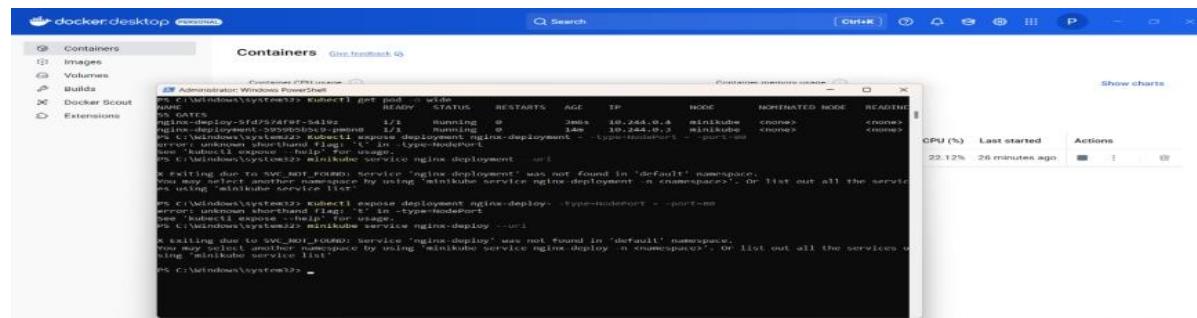
Kubectl expose deployment nginx-deployment - --type=NodePort - --port=80



```
PS C:\Windows\system32> kubectl expose deployment nginx-deployment - --type=NodePort - --port=80
error: unknown shorthand flag: 't' in --type=NodePort
See 'kubectl expose --help' for usage.
PS C:\Windows\system32> minikube service nginx-deployment --url
X Exiting due to SVC_NOT_FOUND: Service 'nginx-deployment' was not found in 'default' namespace.
You may select another namespace by using 'minikube service nginx-deployment -n <namespace>', or list out all the services using 'minikube service list'
PS C:\Windows\system32> kubectl expose deployment nginx-deployment - --type=NodePort - --port=80
error: unknown shorthand flag: 't' in --type=NodePort
See 'kubectl expose --help' for usage.
PS C:\Windows\system32> minikube service nginx-deployment --url
X Exiting due to SVC_NOT_FOUND: Service 'nginx-deployment' was not found in 'default' namespace.
You may select another namespace by using 'minikube service nginx-deployment -n <namespace>', or list out all the services using 'minikube service list'
PS C:\Windows\system32>
```

## 7. To get URL to access the nginx home page

minikube service nginx-deployment --url



```
PS C:\Windows\system32> minikube service nginx-deployment --url
X Exiting due to SVC_NOT_FOUND: Service 'nginx-deployment' was not found in 'default' namespace.
You may select another namespace by using 'minikube service nginx-deployment -n <namespace>', or list out all the services using 'minikube service list'
PS C:\Windows\system32> kubectl expose deployment nginx-deployment - --type=NodePort - --port=80
error: unknown shorthand flag: 't' in --type=NodePort
See 'kubectl expose --help' for usage.
PS C:\Windows\system32> minikube service nginx-deployment --url
X Exiting due to SVC_NOT_FOUND: Service 'nginx-deployment' was not found in 'default' namespace.
You may select another namespace by using 'minikube service nginx-deployment -n <namespace>', or list out all the services using 'minikube service list'
PS C:\Windows\system32>
```

## 8. Copy the above http path and paste it in browser to access it

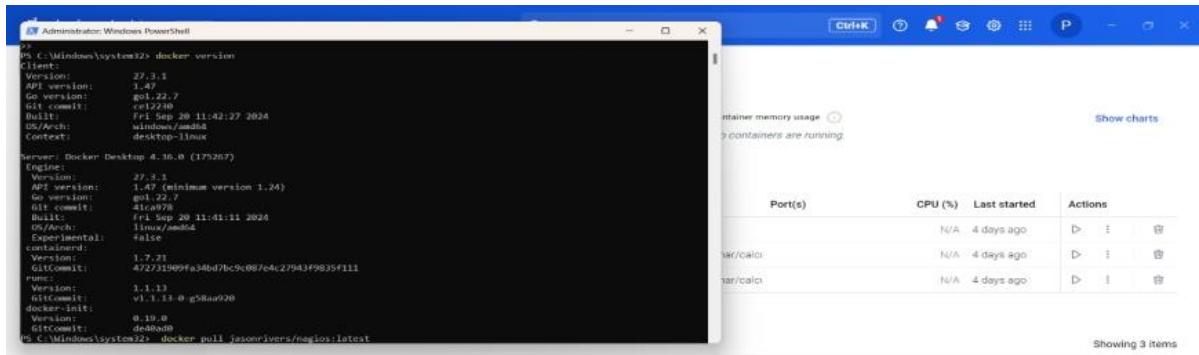


# Experiment – 7G

## Nagios

1.Run Docker Desktop 2.Open PowerShell as administrator and type the following command

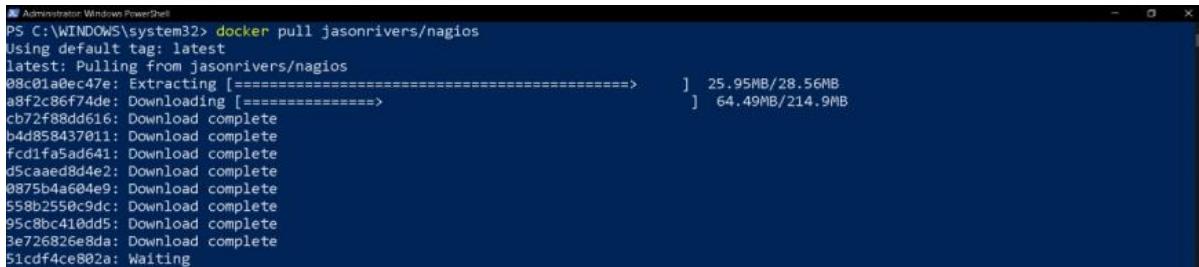
**docker version** [to check that both client and dockere daemon are running]



```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker version
Client:
 Version: 27.3.1
 API version: 1.47
 Go version: go1.22.7
 Git commit: ce12240
 Built: 2024-09-15T20:11:42Z+00:00
 OS/Arch: windows/amd64
 Context: desktop-linux
Server: Docker Desktop 4.16.0 (175267)
Engine:
 Version: 27.3.1
 Engine version: (minimum version 1.24)
 Go version: go1.22.7
 Git commit: 41e0978
 Built: 2024-09-15T20:11:41Z+00:00
 OS/Arch: linux/amd64
 Experimental: false
 containerd: 1.7.21
 containerd commit: 472f31909fa34bd7bc9c087c4c3f9835f111
 runc: 1.1.13
 runc commit: v1.1.15-0-g58aa92b
 docker-init: 0.19.0
 docker-init commit: ded48d0
PS C:\Windows\system32> docker pull jasonrivers/nagios:latest
Showing 3 items
```

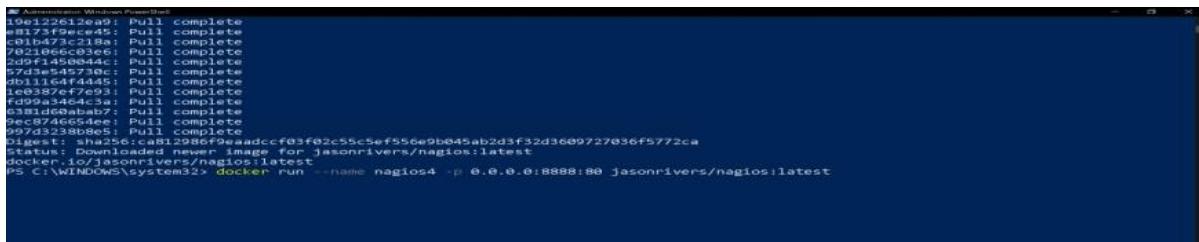
Pull the Nagios image from the Docker Hub with the following command

**docker pull jasonrivers/nagios:latest**



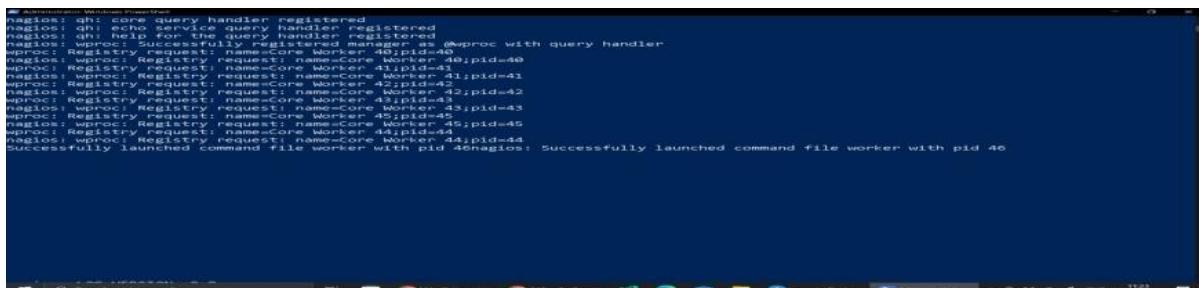
```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker pull jasonrivers/nagios
Using default tag: latest
latest: Pulling from jasonrivers/nagios
08c01a0ec47e: Extracting [=====] 25.95MB/28.56MB
a8fc2c86f74de: Downloading [=====] 64.49MB/214.9MB
cb72f88dd616: Download complete
b4d858437011: Download complete
fd1f1a5ad641: Download complete
d5caeed8d4e2: Download complete
0875b4a604e9: Download complete
558b2550c9dc: Download complete
95c8bc4100d5: Download complete
3e726826e8da: Download complete
51cd4ce802a: Waiting
```

**docker run --name nagios4 -p 0.0.0.0:8888:80 jasonrivers/nagios:latest**



```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker run --name nagios4 -p 0.0.0.0:8888:80 jasonrivers/nagios:latest
Successfully created container 'nagios4' with ID 46
Status: Downloaded newer image for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest
PS C:\Windows\system32>
```

Nagios is successfully launched



```
nagios: ghi: command query handler registered
nagios: ghi: echo service query handler registered
nagios: ghi: help for the query handler registered
nagios: wproc: successfully registered wproc with query handler
wproc: Registry request: name=Core Worker 40,pid=40
wproc: Registry request: name=Core Worker 41,pid=41
wproc: Registry request: name=Core Worker 42,pid=42
nagios: wproc: Registry request: name=Core Worker 43,pid=43
nagios: wproc: Registry request: name=Core Worker 44,pid=44
nagios: wproc: Registry request: name=Core Worker 45,pid=45
nagios: wproc: Registry request: name=Core Worker 46,pid=46
nagios: wproc: Registry request: name=Core Worker 47,pid=47
Successfully launched command file worker with pid 46nagios: Successfully launched command file worker with pid 46
```

We can access Nagios at the browser at port 8888 [localhost:8888]The default credentials for the web interface is nagiosadmin / nagios



Nagios dashboard is the interface to monitor the services

## Nagios®

- General**
- [Home](#)
- [Documentation](#)
  
- Current Status**
- [Tactical Overview](#)
- [Regional \(N-Panel\)](#)
- [Maps](#)
- [Services](#)
- [Host Groups](#)
- [Summary](#)
- [Logs](#)
  
- Service Groups**
- [Summary](#)
- [Logs](#)
  
- Problems**
- [Services \(Pending\)](#)
- [Hosts \(Pending\)](#)
- [Network Outages](#)
- [Quick Search](#)

**Reports**

- [Availability](#)
- [Trends \(Legacy\)](#)
- [Alerts](#)
- [Logs](#)
- [History](#)
- [Metrics](#)
- [Histogram \(Legacy\)](#)
- [Notifications](#)
- [Event Log](#)

**Systems**

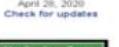
- [Comments](#)
- [Downtime](#)
- [Process Info](#)
- [Performance Info](#)
- [Scheduling Queue](#)
- [Configuration](#)



**Nagios® Core™**  
Version 4.4.6  
April 28, 2020  
Check for updates



**Nagios XI**  
Easy Configuration  
Advanced Reporting  
[Download](#)



**Nagios Log Server**  
Monitor and analyze logs from anywhere  
[Download](#)



**Nagios Network Analyzer**  
Real-time netflow and bandwidth analysis  
[Download](#)

**Get Started**

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of add-ons
- Get support
- Get training
- Get certified

**Latest News**

- Nagios Update - 4.5.6.6
- Nagios Update - 4.5.6.5
- Nagios Update - 4.5.6.4
- More news...

**Quick Links**

- [Nagios Library \(tutorials and docs\)](#)
- [Nagios Labs \(development blog\)](#)
- [Nagios Plugins \(scripts and add-ons\)](#)
- [Nagios Support \(tech support\)](#)
- [Nagios.com \(company\)](#)
- [Nagios.org \(project\)](#)

**Don't Miss...**

- [Monitoring Log Data with Nagios](#) - Nagios Log Server can handle all log data in a variety of formats.
- [Can Nagios monitor netflow?](#) - Yes! Nagios Network Analyzer can take in a variety of flow data. Learn More
- [Nagios XI 5 Available Now!](#) - Easier configuration, Advanced Reporting, Download Today!

Host services [ left side pane can be used for selection.

Click on localhost to check the Host State information

Check on localhost to check the Host State information

The screenshot shows the Nagios interface with the following details:

- Host Status Details For All Host Groups**
- Hosts:** 100 (1 up, 99 down)
- Status:** Up
- Last Check:** 08/04/2012 20:44:44
- Downtime:** Not set (0m 0s)
- Status Information:** Ping OK - Percentage = 0% (0.00ms)

On the left sidebar, the "Current Status" section is expanded, showing the following hosts and their states:

- Host1: Down (since 08/04/2012 20:44:44)
- Host2: Down (since 08/04/2012 20:44:44)
- Host3: Down (since 08/04/2012 20:44:44)
- Host4: Down (since 08/04/2012 20:44:44)
- Host5: Down (since 08/04/2012 20:44:44)
- Host6: Down (since 08/04/2012 20:44:44)
- Host7: Down (since 08/04/2012 20:44:44)
- Host8: Down (since 08/04/2012 20:44:44)
- Host9: Down (since 08/04/2012 20:44:44)
- Host10: Down (since 08/04/2012 20:44:44)
- Host11: Down (since 08/04/2012 20:44:44)
- Host12: Down (since 08/04/2012 20:44:44)
- Host13: Down (since 08/04/2012 20:44:44)
- Host14: Down (since 08/04/2012 20:44:44)
- Host15: Down (since 08/04/2012 20:44:44)
- Host16: Down (since 08/04/2012 20:44:44)
- Host17: Down (since 08/04/2012 20:44:44)
- Host18: Down (since 08/04/2012 20:44:44)
- Host19: Down (since 08/04/2012 20:44:44)
- Host20: Down (since 08/04/2012 20:44:44)
- Host21: Down (since 08/04/2012 20:44:44)
- Host22: Down (since 08/04/2012 20:44:44)
- Host23: Down (since 08/04/2012 20:44:44)
- Host24: Down (since 08/04/2012 20:44:44)
- Host25: Down (since 08/04/2012 20:44:44)
- Host26: Down (since 08/04/2012 20:44:44)
- Host27: Down (since 08/04/2012 20:44:44)
- Host28: Down (since 08/04/2012 20:44:44)
- Host29: Down (since 08/04/2012 20:44:44)
- Host30: Down (since 08/04/2012 20:44:44)
- Host31: Down (since 08/04/2012 20:44:44)
- Host32: Down (since 08/04/2012 20:44:44)
- Host33: Down (since 08/04/2012 20:44:44)
- Host34: Down (since 08/04/2012 20:44:44)
- Host35: Down (since 08/04/2012 20:44:44)
- Host36: Down (since 08/04/2012 20:44:44)
- Host37: Down (since 08/04/2012 20:44:44)
- Host38: Down (since 08/04/2012 20:44:44)
- Host39: Down (since 08/04/2012 20:44:44)
- Host40: Down (since 08/04/2012 20:44:44)
- Host41: Down (since 08/04/2012 20:44:44)
- Host42: Down (since 08/04/2012 20:44:44)
- Host43: Down (since 08/04/2012 20:44:44)
- Host44: Down (since 08/04/2012 20:44:44)
- Host45: Down (since 08/04/2012 20:44:44)
- Host46: Down (since 08/04/2012 20:44:44)
- Host47: Down (since 08/04/2012 20:44:44)
- Host48: Down (since 08/04/2012 20:44:44)
- Host49: Down (since 08/04/2012 20:44:44)
- Host50: Down (since 08/04/2012 20:44:44)
- Host51: Down (since 08/04/2012 20:44:44)
- Host52: Down (since 08/04/2012 20:44:44)
- Host53: Down (since 08/04/2012 20:44:44)
- Host54: Down (since 08/04/2012 20:44:44)
- Host55: Down (since 08/04/2012 20:44:44)
- Host56: Down (since 08/04/2012 20:44:44)
- Host57: Down (since 08/04/2012 20:44:44)
- Host58: Down (since 08/04/2012 20:44:44)
- Host59: Down (since 08/04/2012 20:44:44)
- Host60: Down (since 08/04/2012 20:44:44)
- Host61: Down (since 08/04/2012 20:44:44)
- Host62: Down (since 08/04/2012 20:44:44)
- Host63: Down (since 08/04/2012 20:44:44)
- Host64: Down (since 08/04/2012 20:44:44)
- Host65: Down (since 08/04/2012 20:44:44)
- Host66: Down (since 08/04/2012 20:44:44)
- Host67: Down (since 08/04/2012 20:44:44)
- Host68: Down (since 08/04/2012 20:44:44)
- Host69: Down (since 08/04/2012 20:44:44)
- Host70: Down (since 08/04/2012 20:44:44)
- Host71: Down (since 08/04/2012 20:44:44)
- Host72: Down (since 08/04/2012 20:44:44)
- Host73: Down (since 08/04/2012 20:44:44)
- Host74: Down (since 08/04/2012 20:44:44)
- Host75: Down (since 08/04/2012 20:44:44)
- Host76: Down (since 08/04/2012 20:44:44)
- Host77: Down (since 08/04/2012 20:44:44)
- Host78: Down (since 08/04/2012 20:44:44)
- Host79: Down (since 08/04/2012 20:44:44)
- Host80: Down (since 08/04/2012 20:44:44)
- Host81: Down (since 08/04/2012 20:44:44)
- Host82: Down (since 08/04/2012 20:44:44)
- Host83: Down (since 08/04/2012 20:44:44)
- Host84: Down (since 08/04/2012 20:44:44)
- Host85: Down (since 08/04/2012 20:44:44)
- Host86: Down (since 08/04/2012 20:44:44)
- Host87: Down (since 08/04/2012 20:44:44)
- Host88: Down (since 08/04/2012 20:44:44)
- Host89: Down (since 08/04/2012 20:44:44)
- Host90: Down (since 08/04/2012 20:44:44)
- Host91: Down (since 08/04/2012 20:44:44)
- Host92: Down (since 08/04/2012 20:44:44)
- Host93: Down (since 08/04/2012 20:44:44)
- Host94: Down (since 08/04/2012 20:44:44)
- Host95: Down (since 08/04/2012 20:44:44)
- Host96: Down (since 08/04/2012 20:44:44)
- Host97: Down (since 08/04/2012 20:44:44)
- Host98: Down (since 08/04/2012 20:44:44)
- Host99: Down (since 08/04/2012 20:44:44)
- Host100: Down (since 08/04/2012 20:44:44)

Click on localhost to check the Host State information

### Nagios®

**Host Information**

Last Checktime: Fri Jun 3 05:57:36 UTC 2022  
 Last check: 2022-06-03 05:57:36  
 Nagios Core™ 4.4.8 - www.nagios.org  
 License: Nagios Core 4.4.8

View Status: [Check This Host](#)  
 View Alert History: [For This Host](#)  
 View Event Log: [For This Host](#)  
 View Log: [For This Host](#)  
 View Host Log: [For This Host](#)  
 View Host Log: [For This Host](#)  
 View Host Log: [For This Host](#)

**Current Status**

Tactical Overview  
 Map (Legacy)  
 Services  
 Host Groups  
 Summary  
 Log

**Service Groups**

Group 1  
 Group 2

**Problems**

Servicing (Unacknowledged)  
 Host Problems  
 Network Outages  
 Host Events

**Reports**

Avg. Monitoring  
 Trends (Legacy)  
 History  
 Log  
 Histogram (Legacy)  
 Host Events  
 Event Log

**System**

Comments  
 Downloads  
 Helpdesk  
 Performance Data  
 External Links  
 Configuration

**Host State Information**

Host Status:	Host Last Check:	Host Check Interval:
UP (OK) (Last On Sat 06:04)	2022-06-03 06:04:00	0.00 min
Performance Data:	2022-06-03 06:04:00	3000 (0.000000, 9000, 0.0000, 0.000000, 0.000000)
Current State:	2022-06-03 06:04:00	OK
Last Check Time:	2022-06-03 05:57:36	0.54 min
Check Interval:	2022-06-03 05:57:36	0.54 min
Check Latency / Duration:	2022-06-03 05:57:36	0.54 min
Last State Change:	2022-06-03 05:57:36	OK
Last Modification:	2022-06-03 05:57:36	None (initial alarm state)
Is Host Flapping?	NO	(0.00% flapping)
Last Flapped:	2022-06-03 05:57:36	(0.00% flapping)
Last Scheduled Downtime:	2022-06-03 05:57:36	(0.00% flapping)
Last Uncheduled Downtime:	2022-06-03 05:57:36	(0.00% flapping)

**Active Checks:** ENABLED

Passive Checks:	Event Handler:	Flap Detection:
ENABLED	ENABLED	ENABLED
Observation:	Event Handler:	Flap Detection:
Notifications:	ENABLED	ENABLED

**Host Commands**

- Schedule host downtime for this host
- Un-schedule the last check of this host
- Submit a check result for this host
- Stop monitoring this host for this host
- Stop monitoring over this host
- Create modifications for this host
- Delete modifications for this host
- Unschedule downtime for all services on this host
- Schedule downtime for all services on this host
- Unschedule downtime for all services on this host
- Schedule a check of all services on this host
- Delete a check of all services on this host
- Create checks of all services on this host
- Create event handler for this host
- Clear host notifications for this host
- Clear flapping state for this host

**Host Comments**

Add a new comment  Update all comments

Entry Time Author Comment Command ID Pernament Type Expires Actions

This host has no comments associated with it.

Click on services to check the various services running **CurrentLoad**, **Current User**, **HTTP**, **PING** etc

**Nagios®**

**Current Network Status**  
Last Updated: Fri Jun 10 06:57:41 UTC 2022  
Updated every 90 seconds  
Nagios version 4.4.6  
Logged in as nagiosadmin  
View Hosts For All Hosts  
View Host Status Details For All Hosts

**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
All Problems	All Types		
1	7		

**Service Status Details For All Hosts**

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	06-03-2022 05:53:08	0d 0h 5m 22s+	1/1	OK - load average: 1.03, 0.98, 0.91
localhost	Current Users	OK	06-03-2022 05:53:01	0d 0h 5m 22s+	1/1	USERS OK - 0 users currently logged in
localhost	HTTP	WARNING	06-03-2022 05:57:34	0d 0h 0m 14s	4/4	HTTP WARNING: HTTP/1.1 401 Unauthorized - 695 bytes in 0.001 second response time
localhost	PING	OK	06-03-2022 05:55:17	0d 0h 5m 22s+	1/1	PING OK - Packet loss = 0%, RTT=0.08 ms
localhost	Root Partition	OK	06-03-2022 05:58:08	0d 0h 9m 22s+	1/1	DISK OK: free space: / (237600 MB (0% used) - 8992K)
localhost	Swap Usage	OK	06-03-2022 05:58:03	0d 0h 5m 22s+	1/1	SWAP OK: 100% free (2048 MB total of 2048 MB)
localhost	Total Processes	OK	06-03-2022 05:57:26	0d 0h 5m 22s+	1/1	PROCS OK: 23 processes with STATE=R

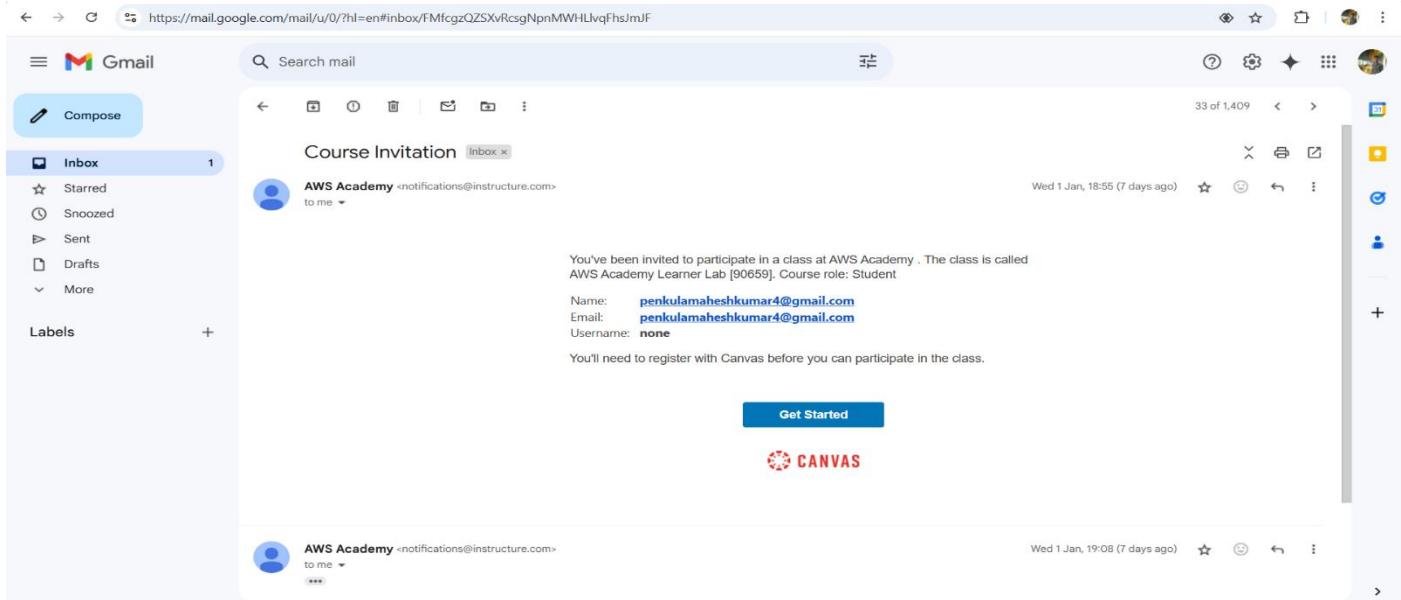
Results 1 - 7 of 7 Matching Services

**Reports**  
Availability Trends (Legacy)

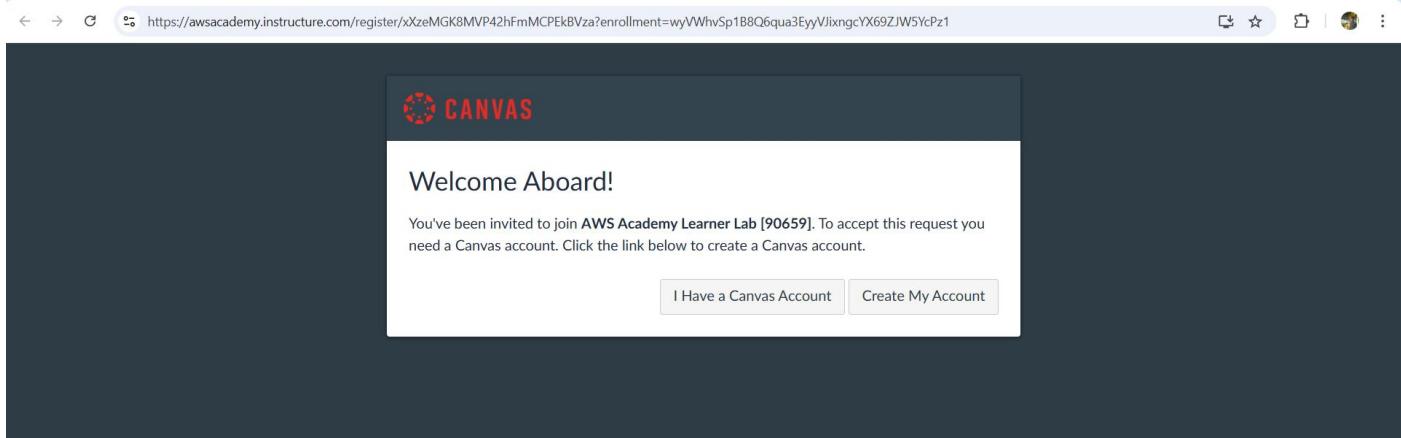
# Experiment – 8A

## AWS ACCOUNT CREATION:

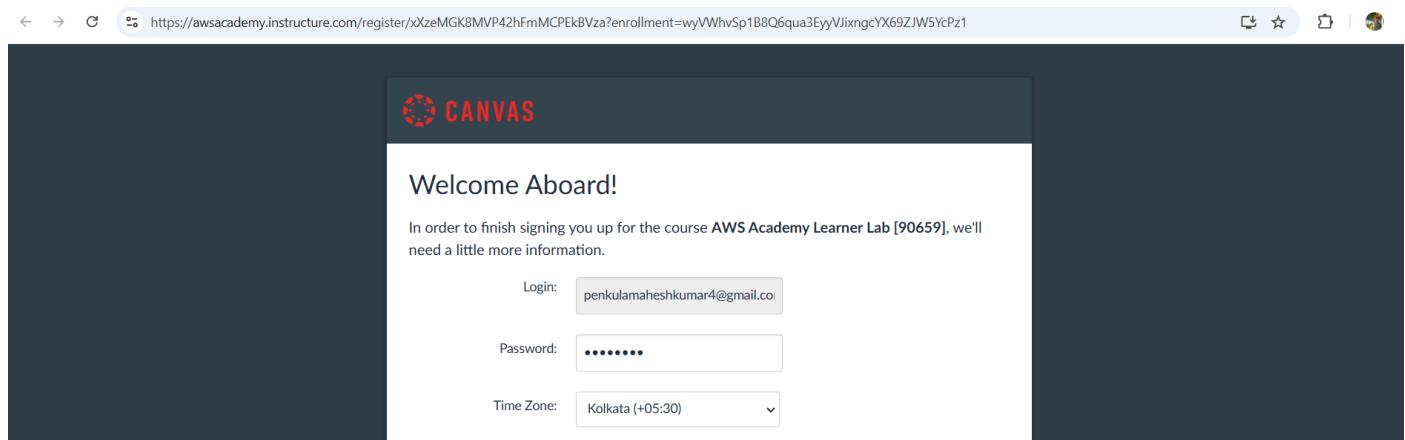
1. Go to your Email and Check for AWS Academy course invitation
2. Click on get started.



3. Click on create my account



4. Provide a password.



5. Select the time zone.check on the boxes.

← → ⌛ https://awsacademy.instructure.com/register/xXzeMGK8MVP42hFmMCPEkBVza?enrollment=wyVWhvSp1B8Q6qua3EyyVJixngcYX69ZJW5YcPz1

The screenshot shows the Canvas registration page. It has a header with the Canvas logo. Below it, a section titled "Welcome Aboard!" displays a message: "In order to finish signing you up for the course AWS Academy Learner Lab [90659], we'll need a little more information." There are three input fields: "Login:" with the value "penkulamaheshkumar4@gmail.co", "Password:" with a masked value, and "Time Zone:" set to "Kolkata (+05:30)".

6. Scroll down and click on register.

← → ⌛ https://awsacademy.instructure.com/register/xXzeMGK8MVP42hFmMCPEkBVza?enrollment=wyVWhvSp1B8Q6qua3EyyVJixngcYX69ZJW5YcPz1

This screenshot shows the same registration page as above, but with two additional checkboxes at the bottom. The first checkbox is labeled "Yes, I'd like Canvas to provide my contact information to Amazon Web Services (AWS) so AWS can share the latest news about AWS services and related offerings with me by email, post or telephone." The second checkbox is labeled "I agree to the Canvas Instructure Acceptable Use Policy and to the AWS Learner Terms and Conditions. The information you provide will be handled by AWS as described in the AWS Privacy Notice." Both checkboxes are checked. A "Register" button is visible at the bottom.

7. On the window click on account.

← → ⌛ https://awsacademy.instructure.com/courses/90659

The screenshot shows the AWS Academy Learner Lab [90659] course page. The left sidebar includes links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area features a title "AWS Academy Learner Lab [90659]" with a decorative background image of clouds and a lightning bolt. Below the title is a paragraph describing the learner lab environment. To the right, there are sections for "View Course Stream", "View Course Calendar", and "View Course Notifications". The "To Do" and "Recent Feedback" sections both show "Nothing for now".

8. Login and logout.

← → ⌛ https://awsacademy.instructure.com/courses/90659

This screenshot shows the same course page as above, but with a user profile overlay in the top left corner. The profile picture is "penkulamaheshkumar4@gmail.com" and there is a "Logout" link. The rest of the page content is identical to the previous screenshot, including the course title, description, and sidebar links.

9. Search for untitled and click.

10. Click on student login

aws academy login

About 674,000 search results

awsacademy.instructure.com › login › canvas  
**Canvas Login | Instructure - AWS Academy**  
Forgot Password? Official Login page for Canvas student login, School Search Canvas, Canvas Network, Canvas Community, and Canvas Free for Teacher accounts.

www.awsacademy.com › vforcesite › LMS\_Login  
**AWS Academy**  
講師の方 (AWS Academyメンバー専用のアカウントをお持ちの方) はこちらからログインしてください。教師用在这里登录 (您需使用AWS Academy Portal账户登录)

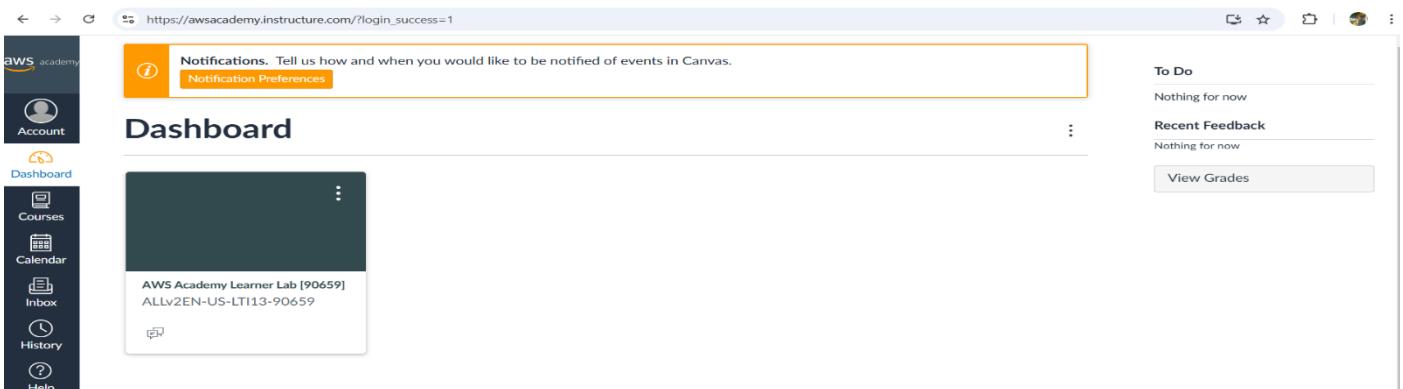
Related searches

- aws training log in
- aws certification log in
- aws academy for students
- aws academy account
- aws student login

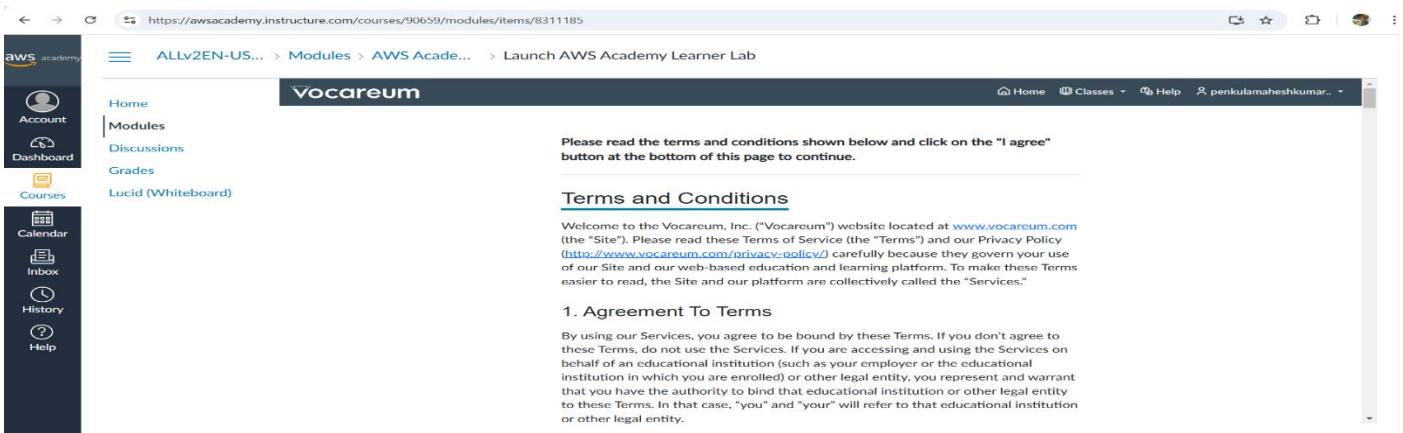
11. Fill the username and password click on login.



12. Click on the aws academy learner lab.



13. Click on modules.



14. Scroll down click on aws academy learner lab.

The screenshot shows the AWS Academy interface. On the left, there's a sidebar with icons for Home, Modules, Discussions, Grades, Lucid (Whiteboard), Courses, Calendar, Inbox, and History. The main content area shows the 'AWS Academy Learner Lab Compliance and Security' module. It has a 'Module Knowledge Check' section with a score of 100 pts and a requirement to 'Score at least 70.0'. There's also a 'Launch AWS Academy Learner Lab' button.

## 15. Scroll down after reading the terms and conditions.

The screenshot shows the Vocareum Terms and Conditions page. It features a note asking users to read the terms and conditions and click 'I agree'. Below this, the 'Terms and Conditions' section is titled '1. Agreement To Terms'. It contains text about agreeing to the terms and conditions of service.

## 16. Click on I agree.

The screenshot shows the Vocareum Terms and Conditions page again. The 'I Agree' button is highlighted in red, indicating it should be clicked. The page also contains sections about notices, communications, and contact information.

## 17. If there is red dot beside AWS it means the sandbox (AWS Learning Environment) is in stop state, click on Start Lab.

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has a top navigation bar with tabs: Start Lab, End Lab, AWS Details, Readme, and Reset. Below the navigation is a terminal window showing the command 'aws \_l\_2944523@runner155711:~\$'. To the right of the terminal is a 'Learner Lab' section with a title 'Environment Overview' and 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 'Environment Overview' section, it says 'Instructions last updated: 2024-12-18'.

18. 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

This screenshot is similar to the previous one, but the AWS status icon in the top right corner has turned green, indicating the lab is active. The rest of the interface, including the terminal, learner lab details, and navigation, remains the same.

19. Once the AWS turns green, click on it

This screenshot is identical to the previous ones, showing the AWS Academy Learner Lab interface with a green AWS status icon. The terminal, learner lab details, and navigation are all present.

## 20. You will land to the AWS dashboard

The screenshot shows the AWS Console Home page. On the left, there's a 'Service menu' box with a message about accessing all AWS services. Below it is a 'No recently visited services' section with links to EC2, S3, RDS, and Lambda. In the center, there's an 'Applications' section showing the Region as US East (N. Virginia). At the bottom, there are three cards: 'Welcome to AWS' (Getting started with AWS), 'AWS Health' (Info), and 'Cost and usage' (Info).

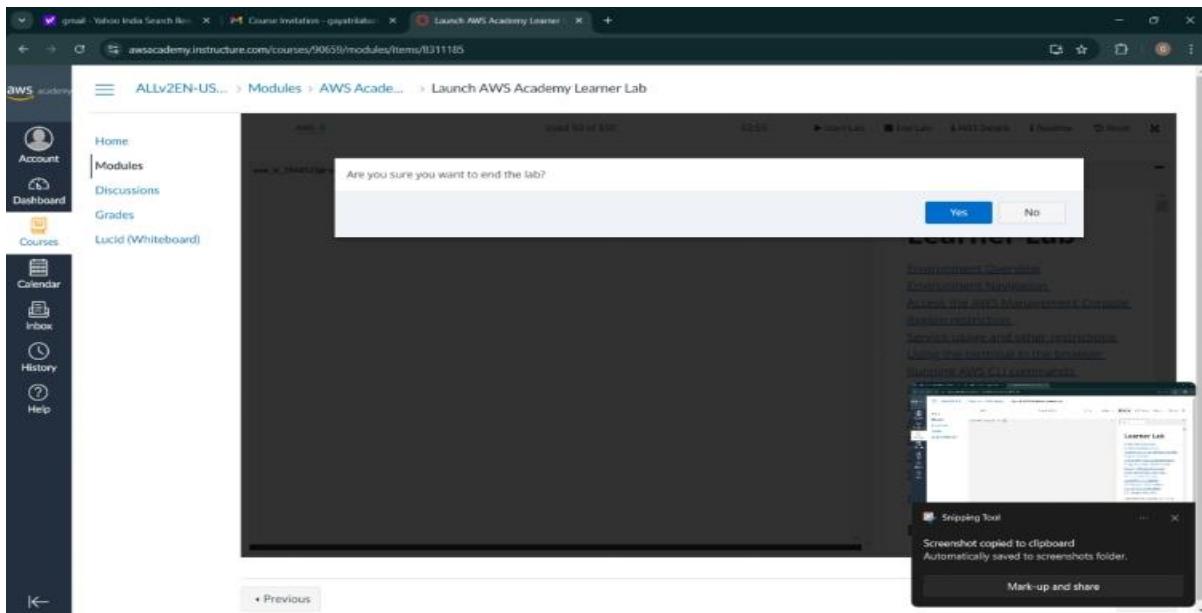
21. Click on EC2 in Recent visited box. We can start the exercises from here

The screenshot shows the AWS EC2 Home page. On the left, there's a sidebar with sections for Instances, Images, Elastic Block Store, Network & Security, and more. The main area shows 'Resources' for the US East (N. Virginia) Region, including 0 instances (running), 0 Auto Scaling Groups, 0 Capacity Reservations, 0 Dedicated Hosts, 0 Elastic IPs, 0 Instances, 1 Key pairs, 1 Load balancers, 0 Placement groups, 1 Security groups, 1 Snapshots, and 0 Volumes. There are also sections for 'Launch instance', 'Service health', 'Zones', and 'Instance alarms'. On the right, there's an 'Account attributes' panel and an 'Explore AWS' panel.

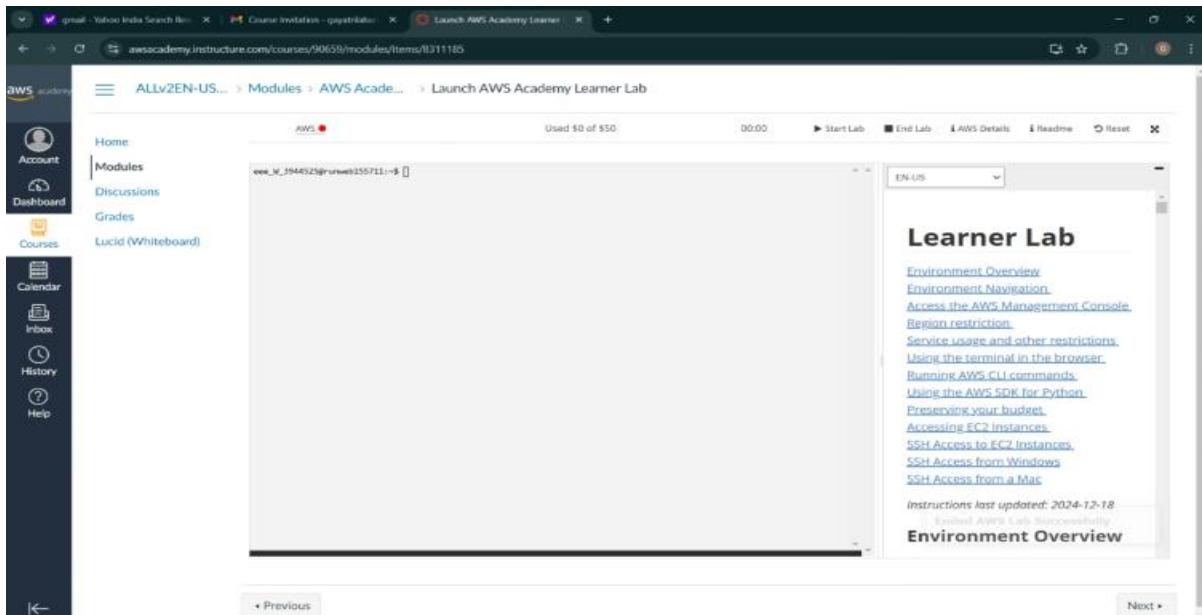
22. Once your lab work is done, click on end lab

The screenshot shows the AWS Academy Learner Lab page. On the left, there's a sidebar with Home, Modules, Discussions, Grades, and Lucid (Whiteboard) options. The main area shows a terminal window with the command 'www\_1\_1944525@runner155711:~\$'. To the right, there's a 'Learner Lab' section with a 'Environment Overview' table and a 'Environment Overview' section with links to various AWS services and tools. Navigation buttons for 'Previous' and 'Next' are at the bottom.

23. Click on yes to end the lab.



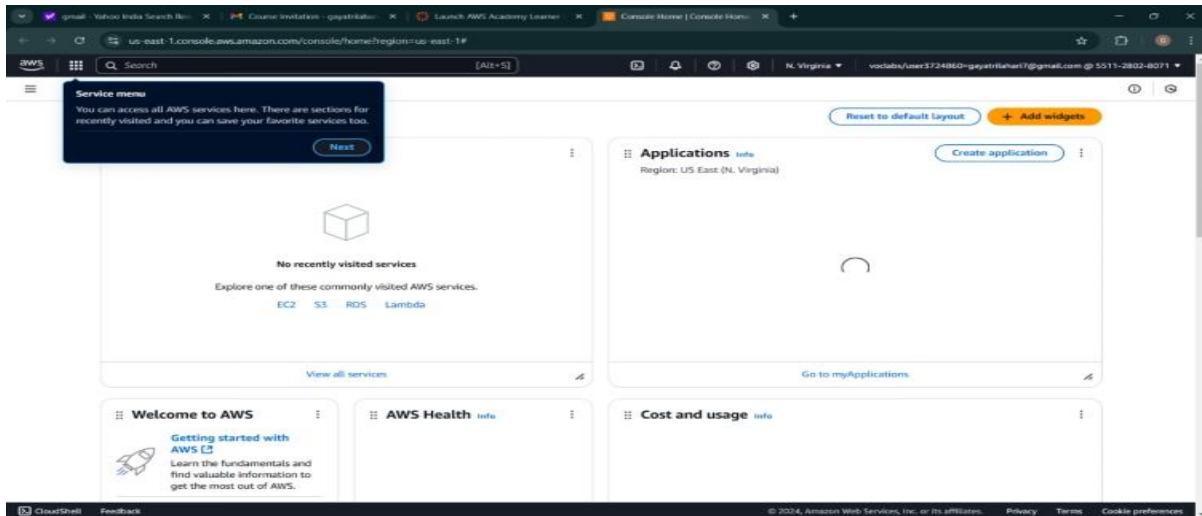
24. Once you get the red color dot beside aws, it indicates stopped



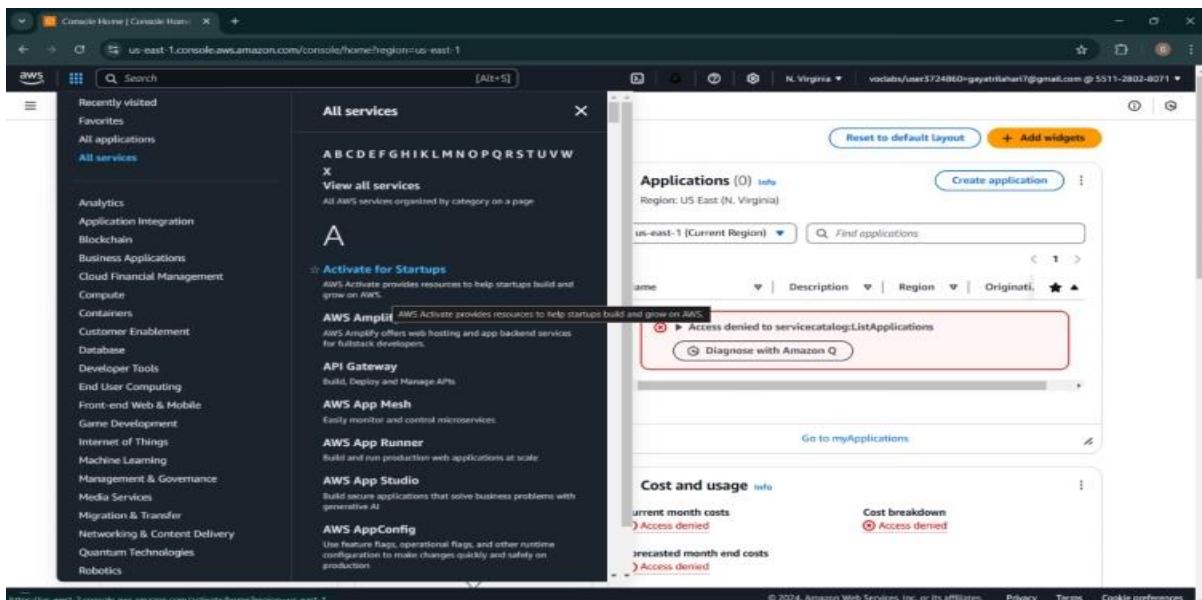
## **Experiment – 8B**

## **DEPLOYING A WEBPAGE USING EC2 INSTANCE ON CLOUD**

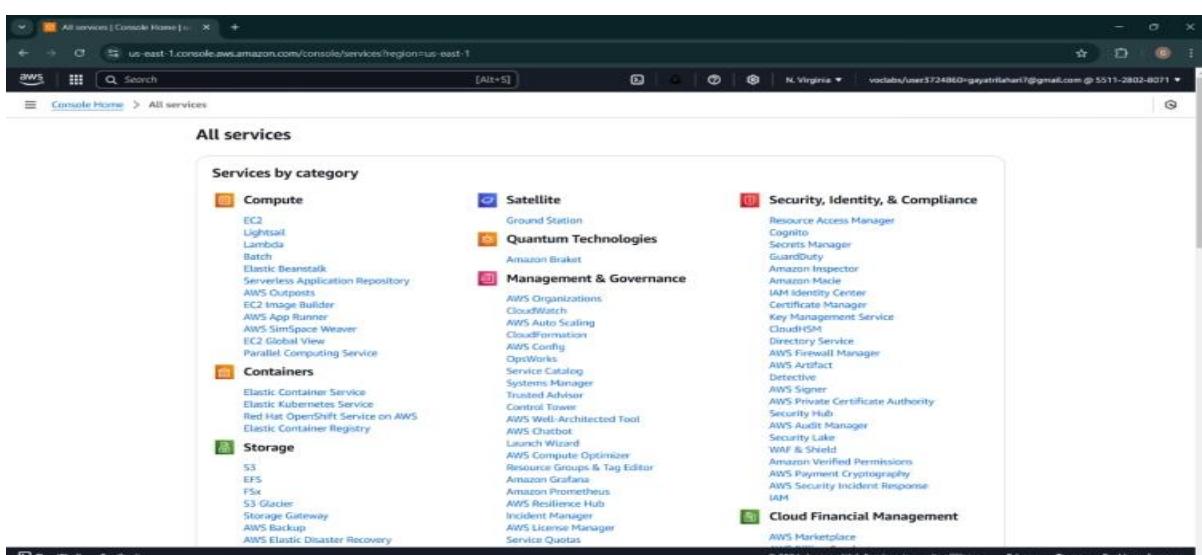
## 1. Enter to the AWS Dashboard



2. Click on Services on top left cornerà then click on All servicesà then click on View all services



3. These are the various services provided by AWS



4. Click on EC2

5. This is the EC2 Dashboard

The screenshot shows the AWS EC2 Global View dashboard. On the left, there's a sidebar with navigation links for Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and Help & Support. The main area has sections for Resources (Instances running: 0, Auto Scaling Groups: 0, Dedicated Hosts: 0, Key pairs: 1, Security groups: 1), Launch instance (Launch Instance button), Service health (AWS Health Dashboard), and Instance alarms (0 in alarm, 0 OK, 0 insufficient data). On the right, there are Account attributes (Default VPC: vpc-06e73fd2558eb640a, Settings, Explore AWS, and a sidebar for Amazon GuardDuty Malware Protection).

6. click Launch instances.

7. This is the Launch an instance dashboard

The screenshot shows the 'Launch an instance' dashboard. It includes a 'Name and tags' section where 'My Web Server' is entered, and a 'Summary' section showing 1 instance. The summary details include Software Image (AMI) as Amazon Linux 2023 AMI 2023.6.2..., Virtual server type (instance type) as t2.micro, Firewall (security group) as New security group, and Storage (volumes) as 1 volume(s) - 8 GB. A tooltip for the free tier is visible, stating: '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 bandwidth to the Internet.'

8. Give a name of your choice to the instance à Scroll down

9. Select the required OS à select Ubuntu

This screenshot is identical to the one above, showing the 'Launch an instance' dashboard. It displays the 'Name and tags' section with 'web1' entered, and the 'Summary' section with the same instance details. A tooltip for the free tier is also present.

10. Under Ubuntu we have many versions à select any Free tier to avoid billing à better go for default

**Summary**

Number of instances: 1

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

Virtual server type (instance type): t2.micro

Free tier: In your first year includes 750 hours of t2.micro for 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 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

**Description**

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

Canonical, Ubuntu, 24.04, amd64 noble image

**Architecture**: 64-bit (x86)

**AMI ID**: ami-0e2c8caa4b6378d8c

**Username**: ubuntu

**Verified provider**

**Launch instance**

11. Select 64-bit (x86) which is default à x86 being a common processor architecture for traditional servers, while Arm is known for its energy-efficient design often used in mobile devices and newer cloud-based systems.

**Summary**

Number of instances: 1

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

Virtual server type (instance type): t2.micro

Free tier: In your first year includes 750 hours of t2.micro for 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 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

**Description**

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

Canonical, Ubuntu, 24.04, amd64 noble image

**Architecture**: 64-bit (x86)

**AMI ID**: ami-0e2c8caa4b6378d8c

**Username**: ubuntu

**Verified provider**

**Instance type**

**t2.micro**

Family: t2

1 vCPU | 1 GB Memory | Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

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

On-Demand SUSE base pricing: 0.0116 USD per Hour | On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

**Additional costs apply for AMIs with pre-installed software**

**Key pair (login)**

**CloudShell** **Feedback**

12. Next is Instance Type à select default à use Free tier only  
An instance type in AWS defines the hardware of a virtual server, specifying its CPU, memory, storage, and networking capacity to meet different application requirements.

**Summary**

Number of instances: 1

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

Virtual server type (instance type): t2.micro

Free tier: In your first year includes 750 hours of t2.micro for 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 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

**Description**

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

Canonical, Ubuntu, 24.04, amd64 noble image

**Architecture**: 64-bit (x86)

**AMI ID**: ami-0e2c8caa4b6378d8c

**Username**: ubuntu

**Verified provider**

**Instance type**

**t2.micro**

Family: t2

1 vCPU | 1 GB Memory | Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

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

On-Demand SUSE base pricing: 0.0116 USD per Hour | On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

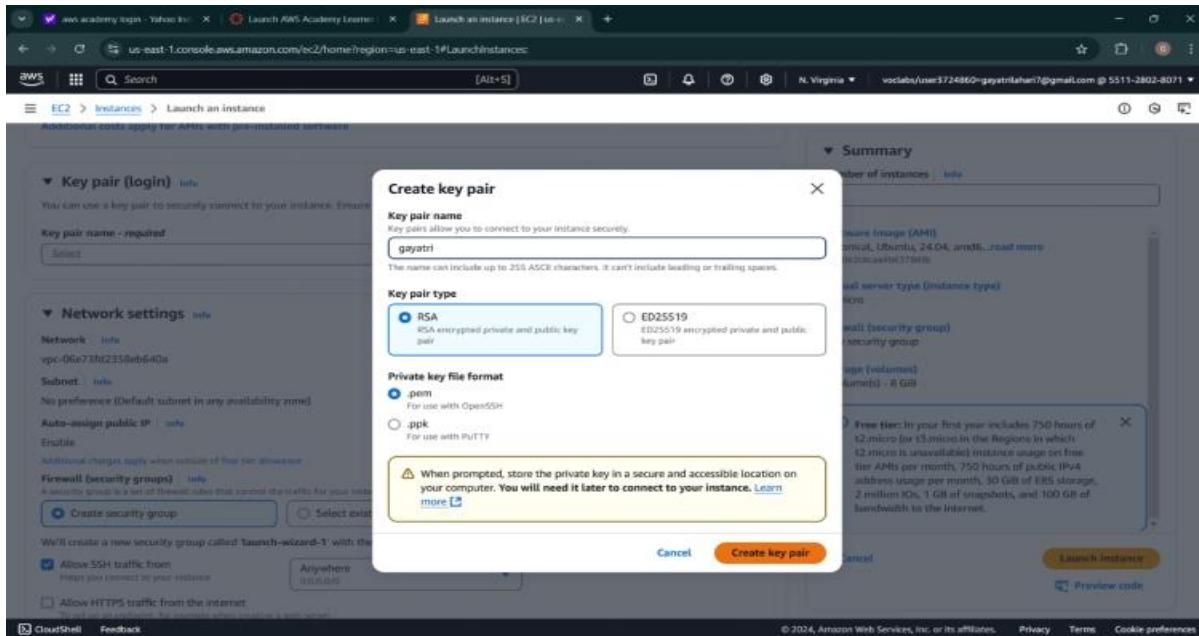
**Additional costs apply for AMIs with pre-installed software**

**Key pair (login)**

**CloudShell** **Feedback**

### 13. Next is Key pair à Click on Create new key pair

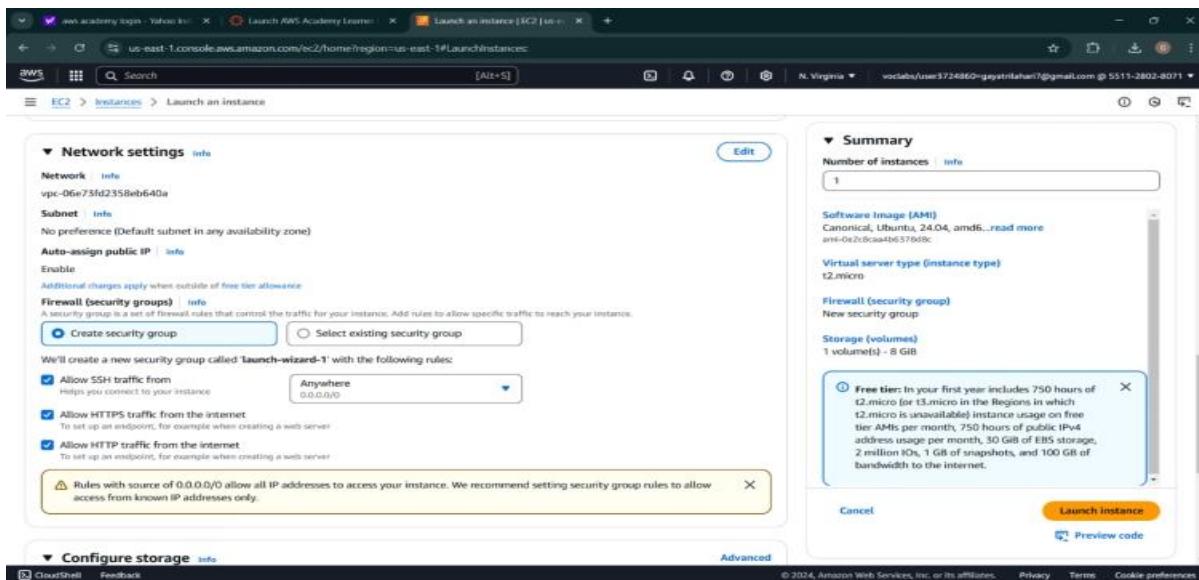
In AWS, a key pair is a set of security credentials, consisting of a public key used to encrypt information and a private key used to decrypt it, allowing secure access to instances and ensuring data protection.



### 14. After the keypair à next is Network settings

### 15. Check the two boxes: Allow HTTPS and Allow HTTP traffic

Allowing HTTPS and HTTP traffic from the internet in AWS network settings enables secure and regular web communication, allowing users to access your web-based applications or websites while ensuring data confidentiality and accessibility.



### 16. Next is the Configuration Storage: Default is 8GB, for Ubuntu we can extend up to 10GB

Configuration storage in AWS refers to the secure and centralized management of settings, preferences, and parameters for various services and applications, ensuring consistent and organized control of your resources.

**Configure storage**

Root volume (Not encrypted)

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

Virtual server type (instance type): t2.micro

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

**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 IOPS, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Launch instance**

17. On the right side, we can see the summary of the instance created à now click on Launch instance in the bottom

**Configure storage**

Root volume (Not encrypted)

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

Virtual server type (instance type): t2.micro

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

**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 IOPS, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Launch instance**

18. The instance is successfully launched à we can either click on instances or scroll down

**Success**

Successfully initiated launch of instance (i-0e782f597dff0632c)

**Next Steps**

- 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
- Manage CloudWatch alarms

19. The instance will start running in few secondsà check the instance state

The screenshot shows the AWS EC2 Instances page with the instance details for i-0e782f397dff632c (web1). The instance is currently running. Key details include:

- Instance ID:** i-0e782f397dff632c
- Public IPv4 address:** 23.22.64.185 | open address
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-26-98.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-06e73fd2358eb640a
- Subnet ID:** subnet-09b085082cd9e59c
- Instance ARN:** arn:aws:ec2:us-east-1:551128028071:instance/i-0e782f397dff632c

20. We will connect using SSH client:

"Connecting to an instance in AWS means establishing a secure and remote access to your virtual server, allowing you to manage and interact with it from your local computer."

21. The command is copied.

The screenshot shows the 'Connect to instance' dialog for the instance i-0e782f397dff632c (web1). The 'SSH client' tab is selected, displaying the following command:

```
ssh -i "gayatri.pem" ubuntu@ec2-23-22-64-185.compute-1.amazonaws.com
```

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

22. Now open PowerShell in Administrator mode

23. Paste the example command copied from the AWS instance

```

Select ubuntu@ip-172-31-26-98:~/awsdemo
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/2005Downloads
PS C:/Users/2005Downloads> ssh -i "gayatri.pem" ubuntu@ec2-23-22-64-185.compute-1.amazonaws.com
The authenticity of host 'ec2-23-22-64-185.compute-1.amazonaws.com (33.22.64.185)' can't be established.
ED25519 key fingerprint is SHA256:kFmSLLTbjcQhdC6Rmf533Opvz5xMv3N6Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-23-22-64-185.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Tue Dec 24 09:30:18 UTC 2024

System load: 0.0 Processes: 184
Usage of /: 24.7% of 6.71GB Users logged in: 0
Memory usage: 28% IPv4 address for enx0: 172.31.26.98
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.

```

Now the Ubuntu is successfully launched into our local system

24. The highlighted is the private IP address to access the instance. GO back to the PowerShell and Update Ubuntu using the highlighted command: sudo apt update

```

Select ubuntu@ip-172-31-26-98:~/awsdemo
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-172-31-26-98:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [572 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [111 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7200 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [795 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [381 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [761 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [173 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [965 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [238 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [309 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [572 kB]
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 B]
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 [948 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]

```

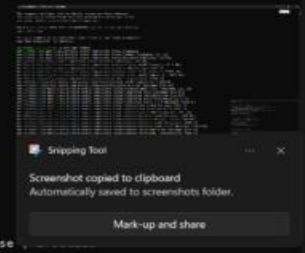
25. After updating Ubuntu install Docker using the command:

sudo apt-get install docker.io

Click Y to continue

26. Install git using the following command:

sudo apt install git

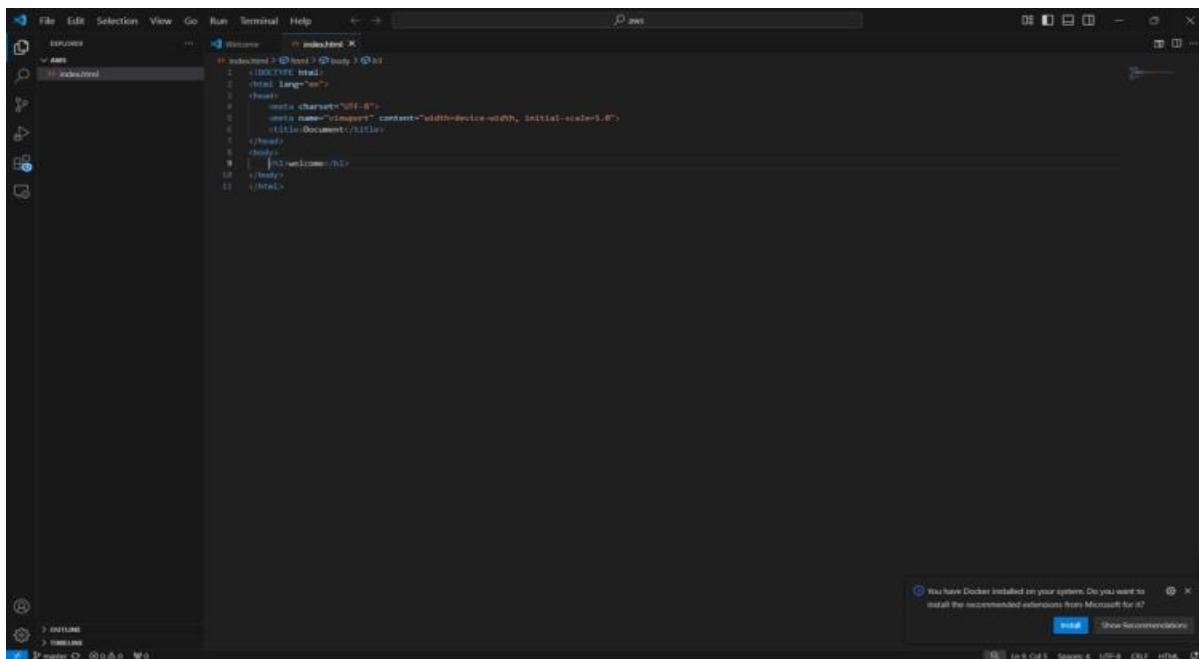


```

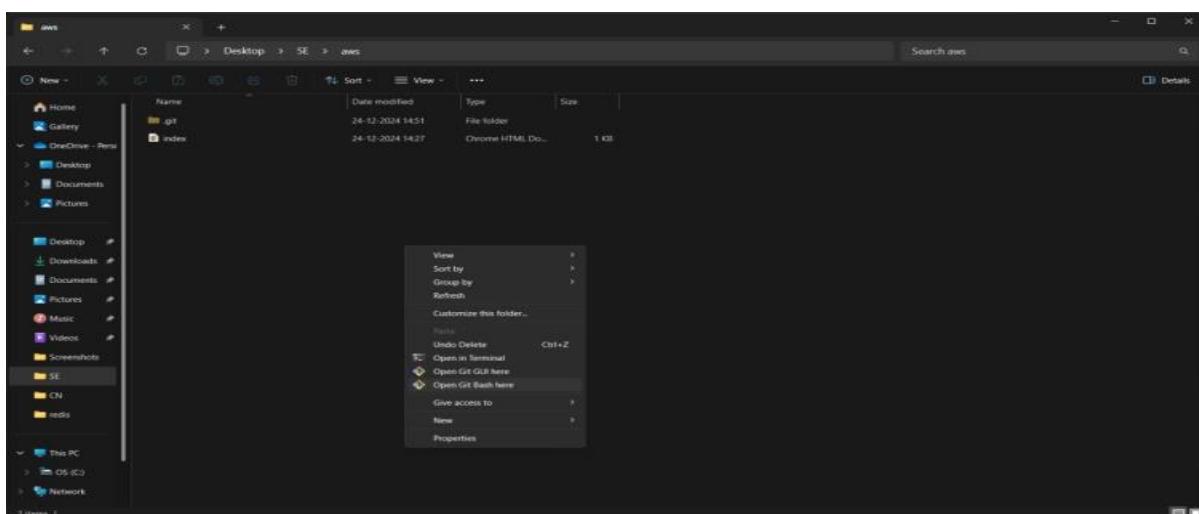
Select ubuntu@ip-172-31-26-98:~/awsdemo
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3844 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu 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]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [111.7 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1084 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 [53.8 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/multiverse amd64 Packages [13.2 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2949 B]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [212 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 31.8 MB in 6s (5427 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-172-31-26-98:~$ 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-172-31-26-98:~$ sudo apt 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-an
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-doc rinse zfs-fuse

```

27. Open any editor like notepad and write a simple web-application using html and save the file as index.html only



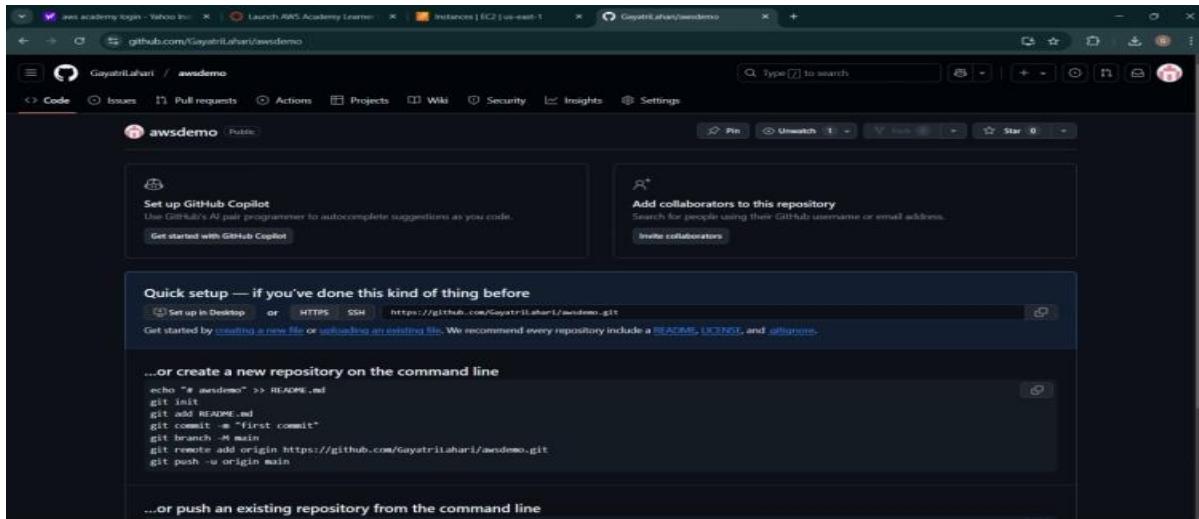
28. Open the folder where your index.html file is located à right click and open GitBash here



29. Type the following Git commands:

```
git init
git add .
git commit
```

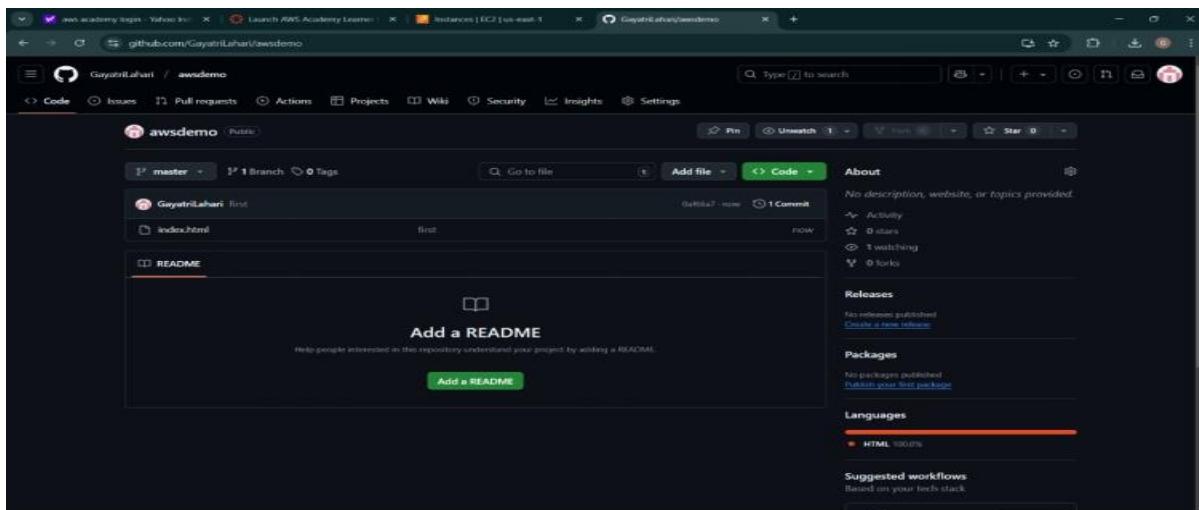
30. Create a repository in GitHub



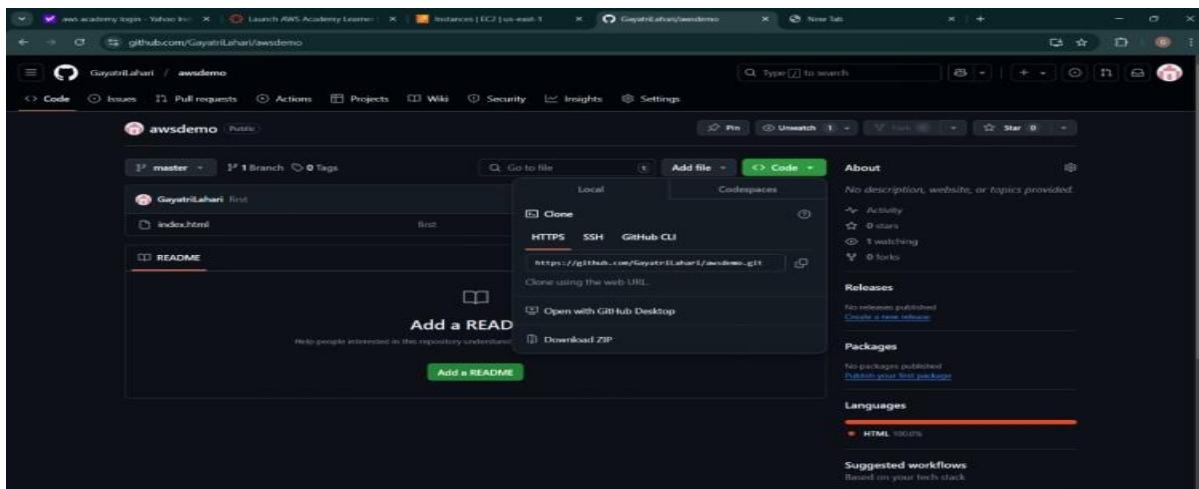
31. Add the remote repository as shown:

```
git remote add origin <paste the HTTPS link of GitHub>.push
```

32. Refresh the GitHub page to see the pushed file.



33. Now copy the HTTPS URL as shown



34. Go back to PowerShell and paste the URL as shown

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-98:~$ git clone https://github.com/GayatriLahari/awsdemo.git
Cloning into 'awsdemo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-26-98:~$ ls
awsdemo
```

35. Once cloning is completedà Move into the folder as shown. Inside this folderà create a Dockerfile as shown

```
Select ubuntu@ip-172-31-26-98:~/awsdemo
Setting up pigr (2.0.1)...
Setting up containerd (1.7.19+really1.7.12-0ubuntu0.1.2) ...
created symlink /etc/systemd/system/multi-user.target.wants/containerd.service + /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service + /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (26.1.3-0ubuntu1-24.04.1) ...
Info: Selecting GID from range 100 to 999 ...
Info: Adding group "docker" (GID 111) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service + /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket + /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-0ubuntu0.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-98:~$ git clone https://github.com/GayatriLahari/awsdemo.git
cloning into 'awsdemo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-26-98:~$ ls
awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ cd awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ nano Dockerfile
Index.html
ubuntu@ip-172-31-26-98:~/awsdemo$ Dockerfile *
```

```
ubuntu@ip-172-31-26-98:~/awsdemo
GNU nano 7.2
FROM nginx
COPY . /usr/share/nginx/html
```

36. Clone the reposiroy.

```
[Select ubuntu@ip-172-31-26-98:~/awsdemo
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.19+really1.7.12-0ubuntu4.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker.io (20.1.3-0ubuntu0.24.04.1) ...
Info: Selecting GID "docker" (GID 113) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Processing triggers for dbus (1.14.10-0ubuntu4.3) ...
Processing triggers for man-db (2.12.0-4build0) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-26-98:~$ git clone https://github.com/GayatriLahari/awsdemo.git
Cloning into 'awsdemo'...
remote: Enumerating objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-26-98:~$ ls
awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ ls
index.html
ubuntu@ip-172-31-26-98:~/awsdemo$ nano Dockerfile
ubuntu@ip-172-31-26-98:~/awsdemo$
```

37. Build a docker image.

38. Nextà Run the Docker container as shown:

```
[Select ubuntu@ip-172-31-26-98:~/awsdemo
To see these additional updates run: apt list --upgradable
1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Tue Dec 24 09:38:20 2024 from 106.220.122.253
ubuntu@ip-172-31-26-98:~$ sudo docker build -t img4
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/
unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/ubuntu/Dockerfile: no such file or directory
ubuntu@ip-172-31-26-98:~$ cd Deckorfile
-bash: cd: Deckorfile: No such file or directory
ubuntu@ip-172-31-26-98:~$ cd awsdemo
ubuntu@ip-172-31-26-98:~/awsdemo$ ls
Dockerfile index.html
ubuntu@ip-172-31-26-98:~/awsdemo$ sudo docker build -t img4 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 63.49kB
Step 1/2 : FROM nginx
latest: Pulling from library/nginx
bc09955b23a84: Pull complete
659ee39b0be5: Pull complete
8cc1c1509e58f5: Pull complete
3862f35d4f001b: Pull complete
13a3280f29cd: Pull complete
7b5839998ea1: Pull complete
57b64962dd94: Pull complete
Digest: sha256:fb197595Sebe76b9C0c14ab68159fd3c08bd067ec62300853543f0ehda353b5be
Status: Downloaded newer image for nginx:latest
---- 66F8Bdd3818c
Step 2/2 : COPY ./ /usr/share/nginx/html
-->> b1b51e99966f2
Successfully built b1b51e99966f2
Successfully tagged img4:latest
ubuntu@ip-172-31-26-98:~/awsdemo$ sudo docker run -d -p 7484:80 img4
8447c64da2184f676a501044e3d9968e85ef3453db1f7d172c9e607a45aaaf
ubuntu@ip-172-31-26-98:~/awsdemo$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8447c64da218 img4 "/docker-entrypoint..." 25 seconds ago Up 24 seconds 8.0.0.7484->80/tcp, ::1:7484->80/tcp youthful_mayer
ubuntu@ip-172-31-26-98:~/awsdemo$
```

39. Now, we can access the web-application using the Public IP address as shownà Copy the address and paste in any browser

**Inbound rules**

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-06bd702fb73850058	SSH	TCP	22	Custom	
sgr-073d5596128a7a27e	HTTPS	TCP	443	Custom	
sgr-0fc9a46a98a9673e	HTTP	TCP	80	Custom	
-	All traffic	All	Custom	Custom	

**Add rule**

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

Cancel Preview changes Save rules

**Instances (1/1) info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
web1	i-0e782f397dff632c	Running	t2.micro	Initializing		us-east-1b	ec2-23-22-1-

**i-0e782f397dff632c (web1)**

- Details**
- Status and alarms
- Monitoring
- Security
- Networking
- Storage
- Tags

**Instance summary**

Instance ID: i-0e782f397dff632c  
 IPv6 address: -  
 Hostname type: IP name: ip-172-31-26-98.ec2.internal

Public IPv4 address: 23.22.64.185 | open address  
 Instance state: Running  
 Private IP DNS name (IPv4 only): ip-172-31-26-98.ec2.internal

Private IPv6 addresses: 172.51.26.98  
 Public IPv4 DNS: ec2-23-22-64-185.compute-1.amazonaws.com | open address

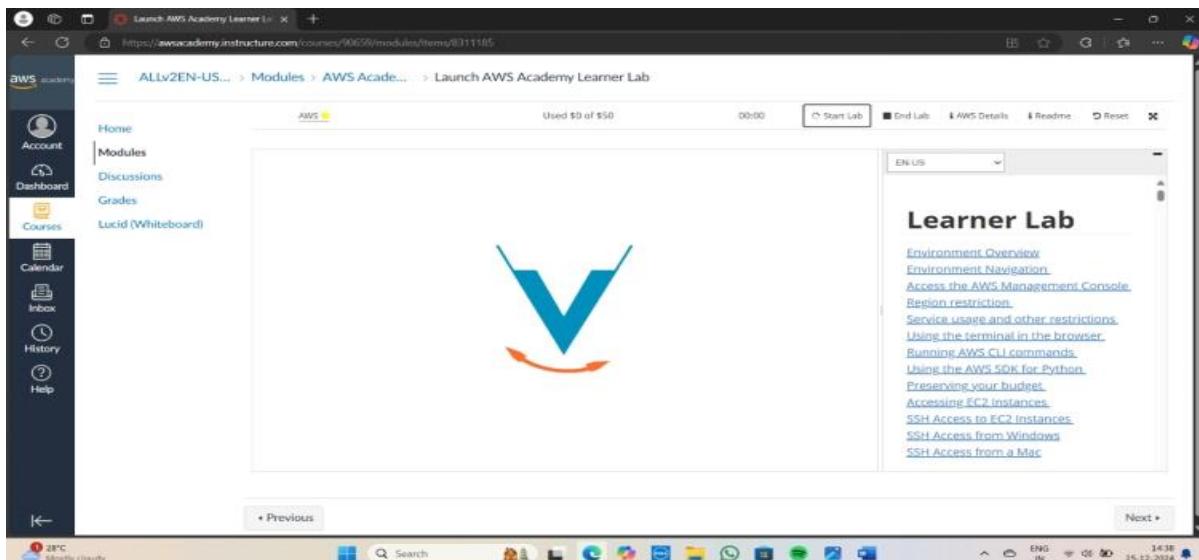
40. We can now see the deployed web application

welcome

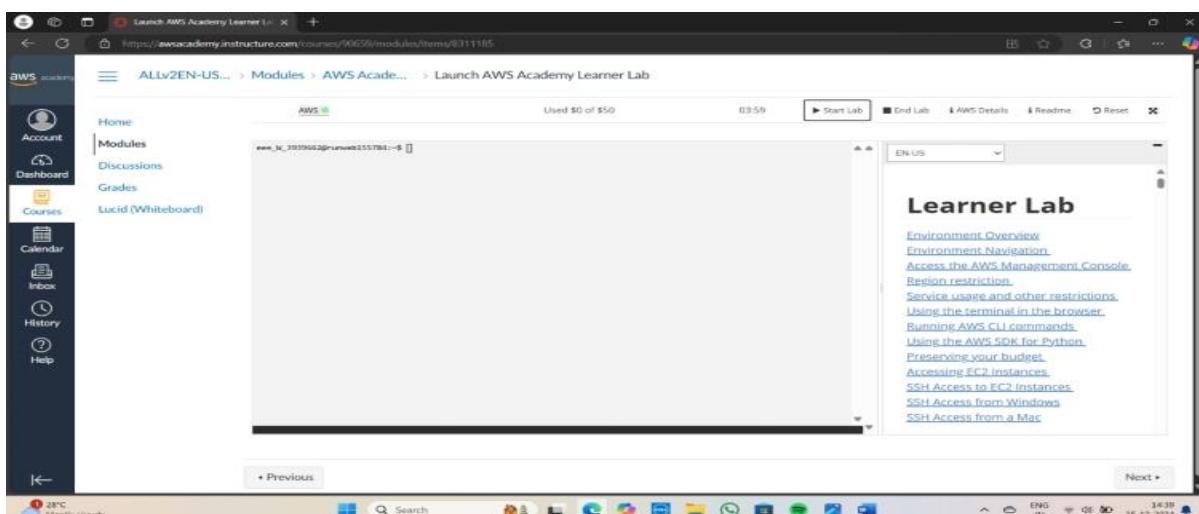
# Experiment – 8C

## MAVEN WEB PROJECT DEPLOYMENT IN AWS

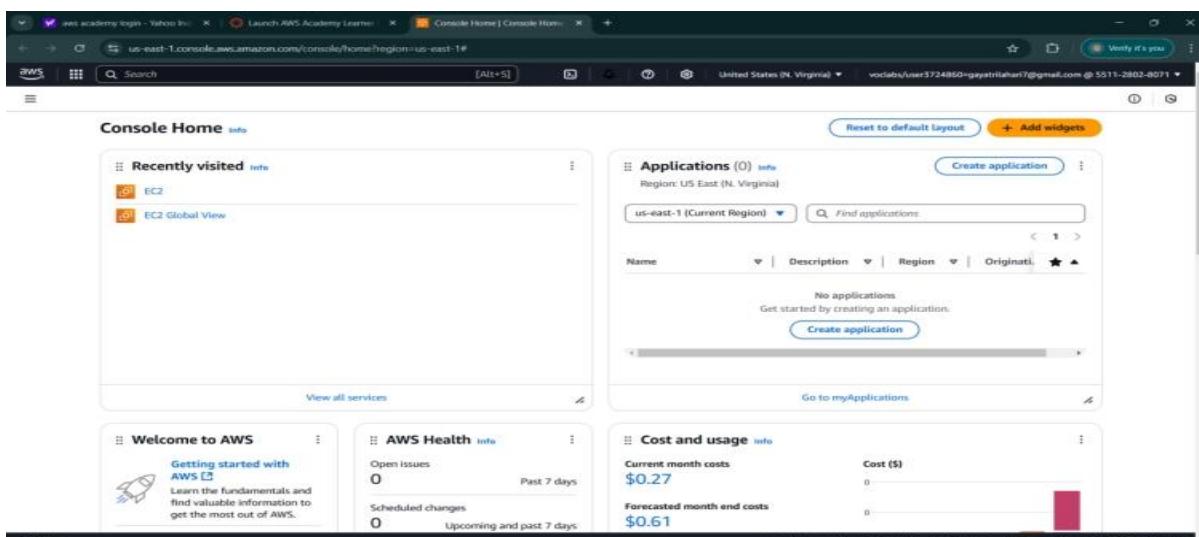
Click on start lab



Click on AWS



Click on EC2



Click on launch instance

Screenshot of the AWS CloudShell interface showing the EC2 Global View dashboard. The dashboard provides an overview of EC2 resources in the US East (N. Virginia) Region, including Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Key pairs, Load balancers, Security groups, and more. It also includes sections for Launch instance, Service health, Zones, and Explore AWS.

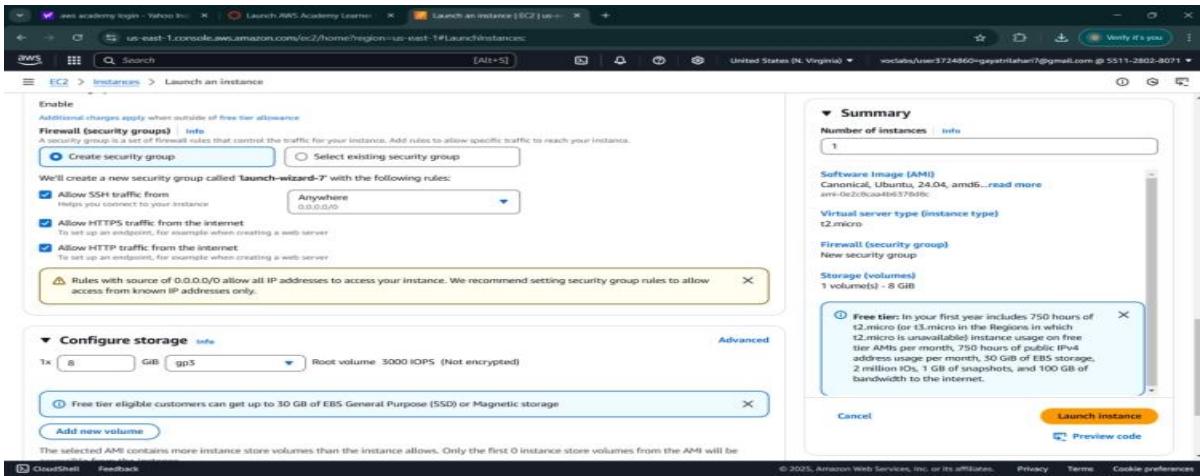
## Give name

Screenshot of the AWS CloudShell interface showing the Launch an instance wizard. The first step, "Launch an instance", shows the configuration for a new instance. It includes fields for Name and tags (Name: MAVENWEBPROJECT), Application and OS Images (Amazon Machine Image) (Ubuntu Server 24.04 LTS (HVM, SSD Volume Type)), and Summary (Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...). A summary box highlights a free tier offer for t2.micro instances.

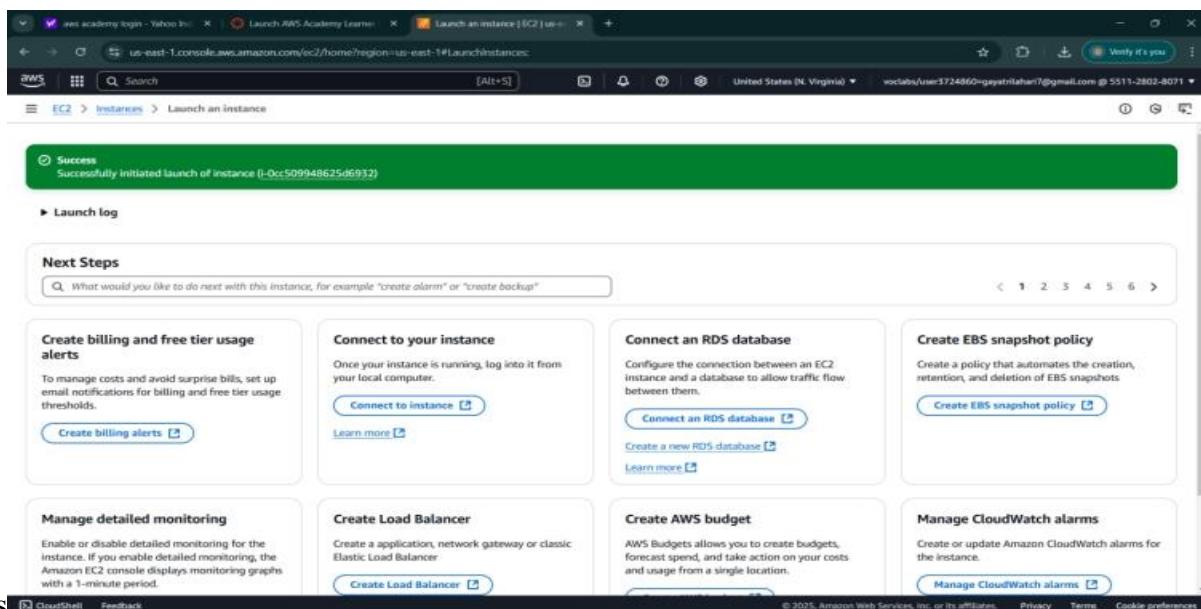
## Create new key

Screenshot of the AWS CloudShell interface showing the "Create key pair" step of the Launch an instance wizard. It shows the creation of a key pair named "examplekey". The "Summary" section remains the same as in the previous screenshot, detailing the software image and instance type. A note at the bottom of the key pair creation form advises users to store the private key securely.

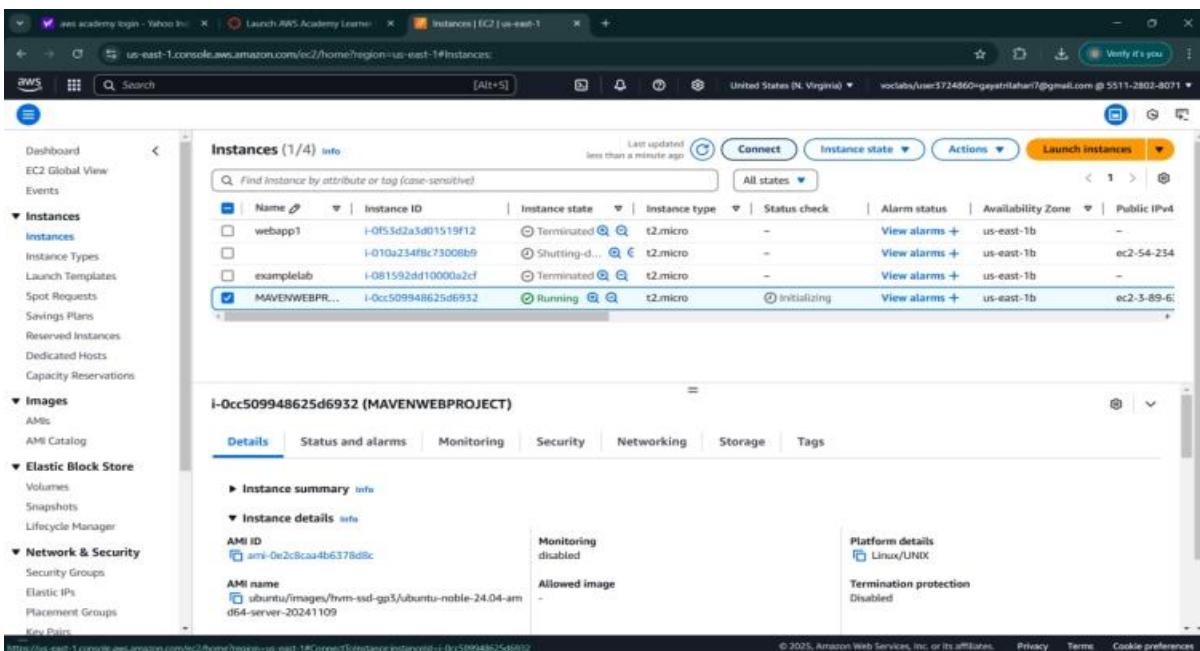
Check all the boxes under network and click on launch instance

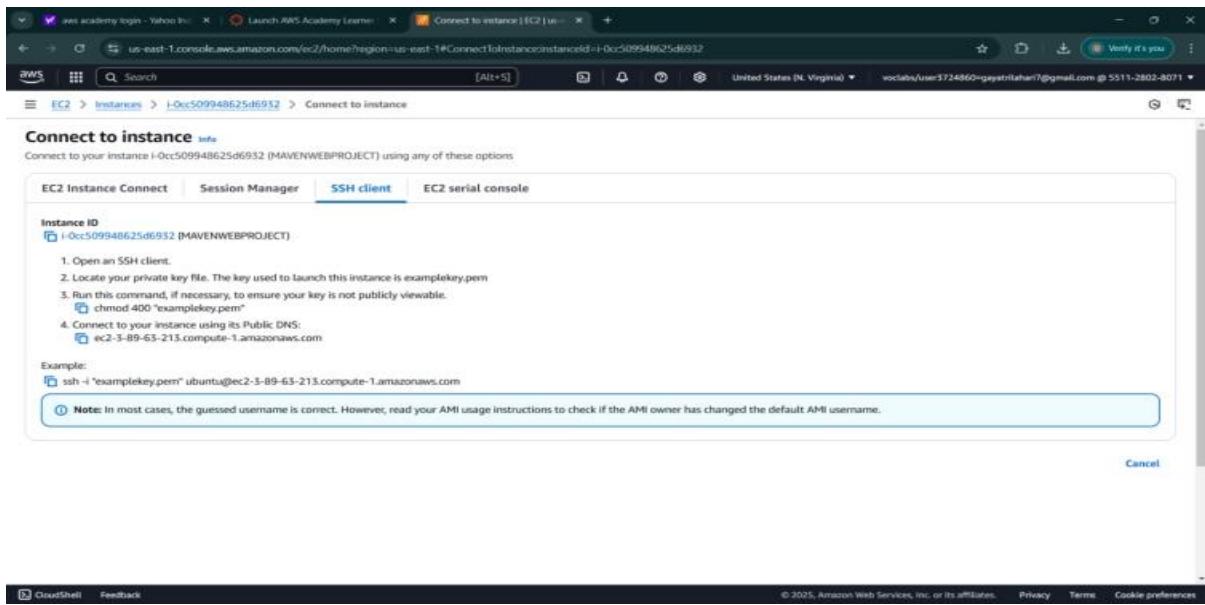


Click on



Wait until you get 2 tests passes and then Click on connect after checking the box





Open the terminal, Navigate to the path

```
C:\Users\palle>cd C:\Users\palle\Desktop\SE_Project
```

Run the SSH command

```
C:\Users\palle\Desktop\SE_Project>ssh -i "Examplekey.pem" ubuntu@ec2-54-152-183-219.compute-1.amazonaws.com
The authenticity of host 'ec2-54-152-183-219.compute-1.amazonaws.com (54.152.183.219)' can't be established.
ED25519 key fingerprint is SHA256:P75YBWfFrD+aX2u4JD0rjANZnxJmH9SeG5bs9blyj0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-54-152-183-219.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Wed Dec 25 15:59:58 UTC 2024

System load: 0.08 Processes: 103
Usage of /: 24.7% of 6.71GB Users logged in: 0
Memory usage: 20% IPv4 address for enx0: 172.31.30.176
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.
```

Run the following commands

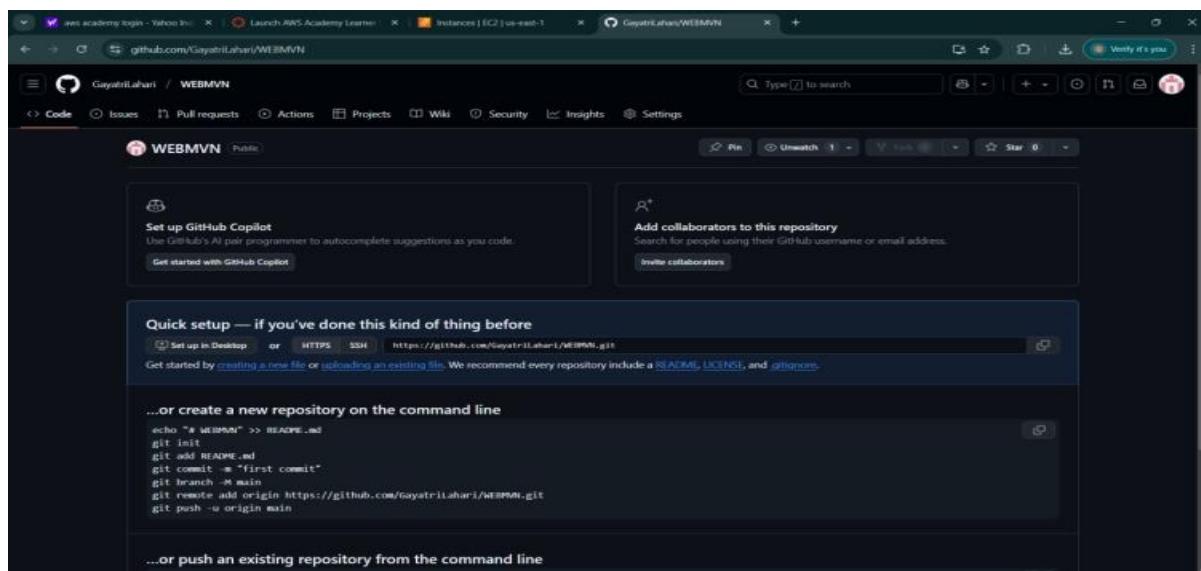
```
ubuntu@ip-172-31-30-176:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [572 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [111 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
```

```
ubuntu@ip-172-31-30-176:~$ 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.
```

```
ubuntu@ip-172-31-30-176:~$ 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-172-31-30-176:~$
```

```
ubuntu@ip-172-31-30-176:~$ 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-172-31-30-176:~$ |
```

Open MAVEN WEB PROJECT REPO in github and copy http link



Clone the repository

```
ubuntu@ip-172-31-30-176:~$ git clone https://github.com/palleAkhila/SampleMavenWebProject.git
Cloning into 'SampleMavenWebProject'...
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.
ubuntu@ip-172-31-30-176:~$ |
```

If your repository is in main run following

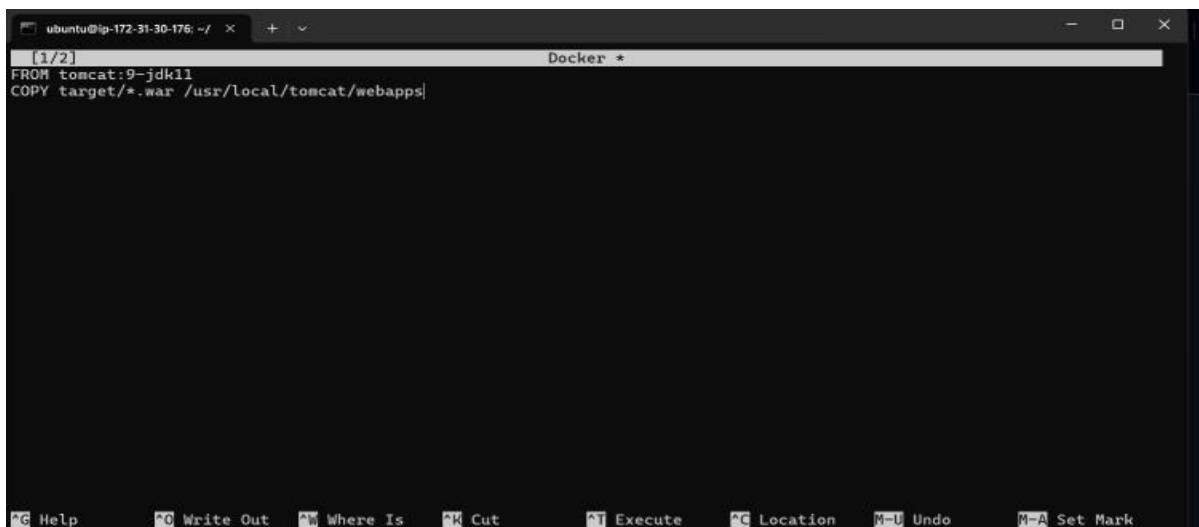
```
ubuntu@ip-172-31-30-176:~$ ls
SampleMavenWebProject
ubuntu@ip-172-31-30-176:~$ cd SampleMavenWebProject/
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ ls
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ |
```

If not goto your github repo

Click on settings

Under default branch section click on the icon shown and you're your master branch as default and execute above command

```
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ nano Dockerfile
```



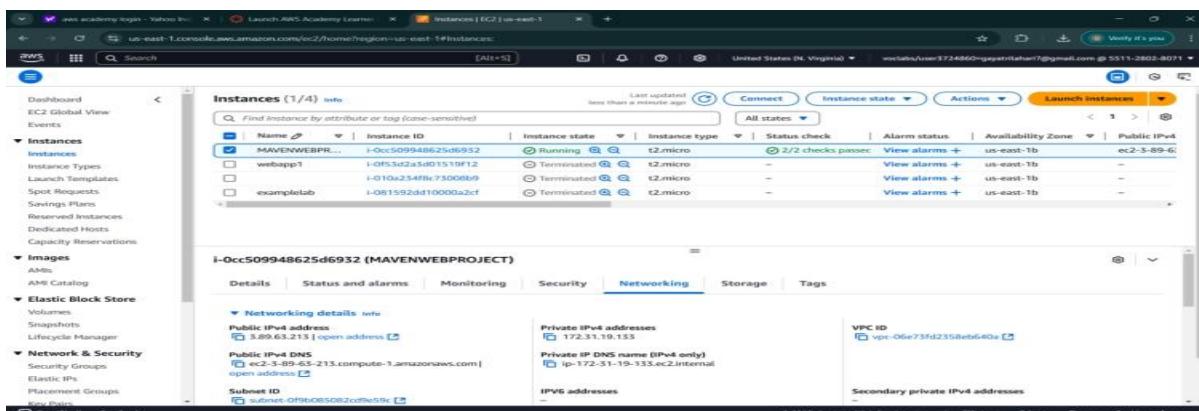
## Build docker image

```
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject/SampleMavenWebProject$ 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 17.92kB
Step 1/2 : FROM tomcat:9-jdk11
--> 1b54e6b2b9a5
Step 2/2 : COPY target/*.war /usr/local/tomcat/webapps/
--> 176e84fb2c8
Successfully built 176e84fb2c8
Successfully tagged mavenwebproject:latest
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject/SampleMavenWebProject$ |
```

## Run the container

```
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject/SampleMavenWebProject$ sudo docker run -d -p 9090:8080 mavenwebproject
e9ffc299c312e259f52fb14e76a5b362ec63b9ed3ed1b86577129cfde1a8b43d
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject/SampleMavenWebProject$ |
```

## Copy public ipv4



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area shows a table of instances. One instance, 'MAVENWEBPROJECT...', is selected and expanded. The 'Networking' tab is selected in the expanded view, showing networking details such as Public IP (3.89.63.213), Private IP (172.31.19.133), and VPC ID (vpc-06e73fd2358eb640a).

Enter this url in browser and click enter

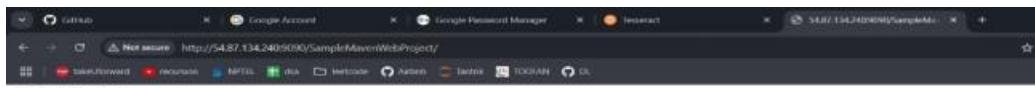


If your app is not running goto security and click on security groups

Click on edit inbound rules

click on add rule give port as 9090 and 0.0.0.0/0 click on save

And refresh the page to check whether the web page has loaded or not



If it is loaded you have successfully deployed your maven web project in your ec2 instance

Run the following commands to stop the container

```
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
e9ffc299c312 mavenwebproject "catalina.sh run" 6 minutes ago Up 6 minutes 0.0.0.0:9090->8080/tcp, :::9090->8080/tcp silly_mirzakhani
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ |
```

```
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ sudo docker stop e9ffc299c312
e9ffc299c312
ubuntu@ip-172-31-30-176:~/SampleMavenWebProject$ |
```

Terminate your instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like 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, Elastic IPs, Placement Groups, Key Pairs, CloudShell, and Feedback.

The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
MAVENWEBPR...	i-Occ509948625d6932	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-89-6-
webapp1	i-0f53d2a3d01519f12	Terminated	t2.micro	-	View alarms +	us-east-1b	-
i-010a234fb7c73000b9	i-010a234fb7c73000b9	Terminated	t2.micro	-	View alarms +	us-east-1b	-
examplelab	i-081592dd10000a2cf	Terminated	t2.micro	-	View alarms +	us-east-1b	-

A context menu is open over the first instance, 'MAVENWEBPR...', with the 'Terminate (delete) instance' option highlighted. Other options in the menu include Stop instance, Reboot instance, and Hibernate instance.

The screenshot shows the AWS EC2 Instances page with a confirmation dialog for terminating the instance 'MAVENWEBPR...'. The dialog contains the following text:

⚠️ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

Instance ID: i-Occ509948625d6932 (MAVENWEBPROJECT) | Termination protection: Disabled

To confirm that you want to delete the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

Cancel | **Terminate (delete)**

End your lab