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

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

## DEPARTMENT OF INFORMATION TECHNOLOGY

# LAB RECORD

# SOFTWARE ENGINEERING LAB

**B. Tech. III YEAR I SEM (KR23)**

**ACADEMIC YEAR 2025-26**

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad
Narayanguda, Hyderabad, Telangana – 500029

# Certificate

This is to certify that following is a Bonafide Record of the workbook task done by _____bearing Roll No_____ of _____ Branch of _____ year B. Tech. Course in the_____ Subject during the Academic year _____ & _____ under our supervision.

Number of week tasks completed: _____

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner

**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**
**(AN AUTONOMOUS INSTITUTE)**
**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad**
**Narayanguda, Hyderabad, Telangana – 500029**

# Daily Laboratory Assessment Sheet

Name of the Lab: Student Name:
Branch & Section: HT. No:

| Sr. No. | Name of the Experiment | Date | Observation Marks (5M) | Record Marks (5M) | Viva Voice Marks(10M) | Total Marks (20M) | Signature of Faculty |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | **Total** | | | | | | |

**Faculty Incharge**

# INDEX

| | | |
|---|---|---|
| | b. Gain skills in writing and interpreting docker-compose.yml files for service setup.<br>c. Deploy the same setup across different machines without manual configuration.<br>d. Configure container networking and persistent storage within Compose.<br>e. Reduce setup time and enable faster iteration during application development. | |
| 7. | **Creating a Multi-Module Maven Project**<br><br>a. Build and package Java and Web applications using Maven.<br>b. Add dependencies using **pom.xml**, compile and test using plugins.<br>c. Resolve errors and conflicts arising from dependency mismatches.<br>d. Work with parent and multi-module Maven projects.<br>e. Generate executable JARs and deployable WARs using Maven | |
| 8. | **Jenkins Automation**<br>a. Hands-on practice on manual creation of Jenkins pipeline using Maven projects from Github<br>b. Create the job and build the pipeline for maven-java and maven-web project. | |
| 9. | **Pipeline Creation using script**<br>a. Evaluation of Jenkins pipeline.<br>b. WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE<br>c. Hands-on practice on creation of scripted Jenkins pipeline.<br>d. Take the screenshots for above task | |
| 10. | **Working with minikube and Nagios**<br>a. Hands-on practice of creating, running and scaling pods in minikube.<br>b. Running Nginx server on specified port number by explaining the Nginx monitoring tool<br>c. Running Nagios server and Understanding the Monitoring tool using Docker.<br>d. AWS-free Trier account Creation steps | |
| 11. | **Jenkins-CI/CD**<br><br>a. CI-Continuous Integration using Webhooks.<br>b. Sending E-mail Notification on Build Failure or success | |
| 12. | **Creation of virtual machine for Ubuntu OS and Deploying the web application**<br>1. Creation of virtual machine<br>2. Deploying the web application<br>3. Accessing it publicly | |

**LAB LESSON PLAN FOR WEEK 1**

Lab Topics:

- Installation of Eclipse and other software (Installation PDF is available and will be shared before lab.)
- Downloading Maven and Configuring

- Student has to prepare on III/I project topic the following:

  - ✓ Abstract
  - ✓ Functional requirement
  - ✓ Non- Functional requirement
  - ✓ Identify user of application
  - ✓ Modules in application
  - ✓ Use case diagram

- **Use case diagram**

Unified Modeling Language (UML) diagrams are a way to visualize systems and software. It helps the software engineers understand designs and proposed implementation of complex software systems. UML (Unified Modeling Language) diagrams are categorized into two main types: **Structural diagrams** and **Behavioral diagrams**. These diagrams serve to model different aspects of a system, from its architecture and components to its behaviour and interactions.

# Software Requirements Specification (SRS):

## Project Title: Gen AI in Education - Personalized Quiz and Content Generator Using LLM.

- **Abstract:** This project develops an AI-based system that uses Large Language Models (LLMs) to generate personalized quizzes and tests from educational content. It adapts questions to each student's learning level and performance, providing instant feedback and explanations. The system helps educators track progress and tailor instruction, enhancing learning outcomes through intelligent assessment.
- **Functional Requirements:**
  a. **User Registration & Authentication:** Students, teachers, and admins can register and log in securely.
  b. **Content Upload:** Teachers can upload educational materials (PDFs, notes, transcripts).
  **c. Content Processing:** The system extracts key concepts from uploaded content using LLM.
  d. **Personalized Quiz Generation:** LLM generates quizzes tailored to each student's learning profile and content.
  e. **Quiz Attempt and Submission:** Students can take quizzes and submit answers through the platform.
  f. **Feedback Generation:** The system provides instant feedback and explanations for each question.

g. **Performance Tracking:** Student scores, accuracy, and progress are stored and visualized.
h. **Teacher Dashboard:** Teachers can view student analytics and quiz content.
i. **Admin Controls:** Admins can manage users, content repositories, and LLM configuration.
j. **LLM API Integration:** The system connects to external LLM services (e.g., OpenAI API) for generational tasks.

- ## Non-Functional Requirements:

1. **Usability:** The interface should be user-friendly and intuitive for all users.
2. **Performance:** Quiz generation time should be under 5 seconds.
3. **Scalability:** The system should support 1,000+ concurrent users.
4. **Security:** All data should be encrypted; authentication must be secure.
5. **Availability:** The system should have at least 99.9% uptime.
6. **Maintainability:** Codebase should be modular and well-documented for future updates.
7. **Compatibility:** Must work on desktops, tablets, and smartphones.

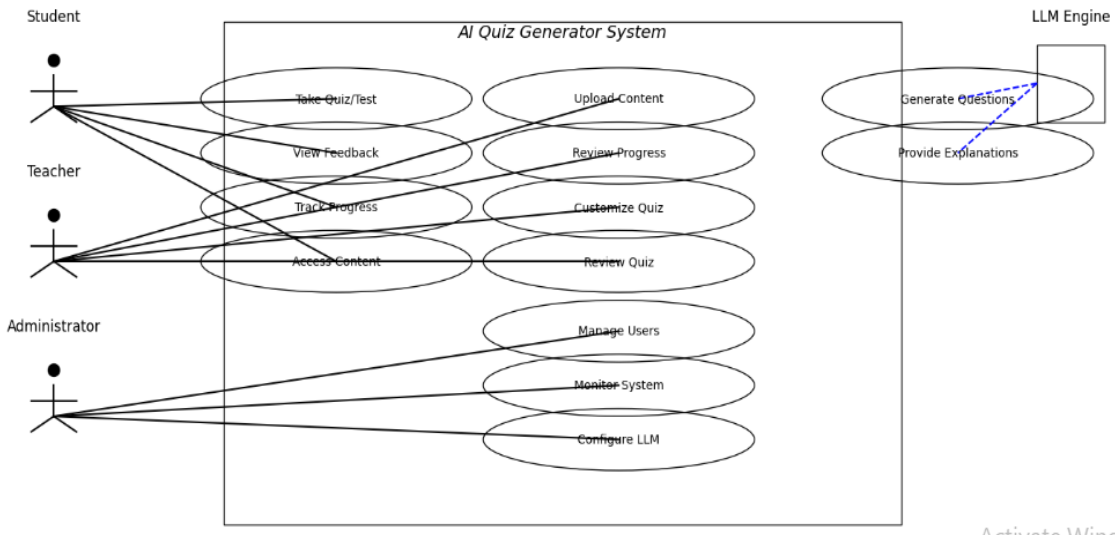
- ## Use Cases:

1. **User Registration and Login:** Users (students, teachers, admins) create accounts and securely log in.
2. **Content Upload:** Teachers upload educational materials such as PDFs, notes, and transcripts.
3. **Content Processing:** The system processes uploaded content to extract key concepts using LLM.
4. **Personalized Quiz Generation:** The system generates quizzes tailored to each student's learning progress and content.
5. **Quiz Attempt:** Students take personalized quizzes within the platform.
6. **Instant Feedback and Explanation:** After quiz submission, students receive AI-generated feedback and detailed explanations.
7. **Performance Tracking:** The system records student quiz scores and tracks progress over time.
8. **Teacher Analytics Dashboard:** Teachers view student performance reports

and quiz results.

9. **Quiz Review and Approval:** Teachers can review and edit AI-generated quizzes before assigning them.
10. **User Management:** Admins manage users, roles, and access permissions.
11. **System and LLM Configuration:** Admins configure LLM settings and integration parameters.

- **Workflow Diagram:**



**Task :** Students must explore all git commands on given scenario-based question. (Note student write command in observation book and paste the answer (git command) below each question for uploading).

**Scenario based questions on basic git commands**

1. You made changes to one file in the above project but haven't staged them yet. You realize they were a mistake. What Git command will you use to discard the local changes?

git checkout -- <filename>

2. You accidentally ran git add file1.txt (note instead of file.txt consider one file from above project), but you're not ready to commit yet. How do you remove it from the staging area without losing the changes?

git reset file1.txt

3. You made a commit but typed the wrong commit message. You haven't pushed it yet. How do you fix it?

git commit --amend

4. You want to view the commit history of the current branch in a readable format. What Git command should you use?

git log --oneline --graph

5. After cloning a repo, how can you set your name and email globally for all Git repositories?

git config --global user.name "Name"

git config --global user.email "Mail Id"

6. You've made some edits to your files but haven't staged them. How can you view the changes?

git diff

7. You're on the main branch but need to switch to feature/login. What command do you

git checkout -b feature -branch

8. You deleted the feature-ui branch by mistake. You haven't pushed the deletion. How

git branch -d feature-branch

9. You've made some commits locally and now want to upload them to the remote repository. What do you run?

git push

10. How can you fetch the latest changes from the remote repository without merging them automatically?

git fetch

11. You're on the main branch and want to start working on a new feature called search-filter. What command do you use?

git checkout -b search-filter

12. You committed a file containing an API key and want to completely remove it from the repo's history. What should you do?

git filter-branch --force

13. You want to see all the branches that exist both locally and on the remote. What

git branch -a

14. You're on main and want to merge the completed feature/signup branch into it. What command do you use?

git merge feature/signup

15. You tried to merge two branches and Git reported a conflict in app.js. What are the general steps to resolve it?

git add app.js

git commit

16. You don't want Git to track changes to .log files or node_modules/. How do you achieve this?

echo "*.log" >>.gitignore

echo "node_modules/".gitignore

17. You're investigating a bug and want to know who last changed line 25 in script.py. What command do you use?

git blame -L 25,25 script.py

18. You're in the middle of working on a file but need to switch branches quickly. What do you do to save your work?

git stash

19. You previously ran git stash and now want to restore those changes. What command do you use?

git stash pop

20. The feature/test branch is no longer needed. How do you delete it locally?

git branch -d feature/test

21. You just merged the feature-ui branch into main. You want to clean up your local branches. How do you safely delete feature-ui?

git reset file1.txt

22. You created a branch feature-experiment, made some changes, but now realize the code is no longer needed. You want to delete it, even though it hasn't been merged. What command will you use?

git branch -D featureexperiment

23. You're currently on the feature-login branch and want to delete feature-ui. What must you ensure before running the delete command?

git checkout main

git branch -d feature-ui

24. You want to delete bugfix-footer, but you're unsure if it's been merged. What should you do before deleting it?

git branch --merged | grep

25. You finished working on feature-a, feature-b, and feature-c. All have been merged into main. How do you delete them in one command?

**Scenario based questions on working with remote repository using Git commands**

1. Specify the git command when you're starting work on a project and need to clone a remote repository to your local machine.

git clone <repo url>

2. Specify the git command if you want to see the remotes that are connected to your local repository.

git remote -v

3. Specify the git command if you want to add a new remote repository to your local repository.

git remote add <name> <url>

4. Specify the git command to remove a remote repository from your local configuration:

git remote remove <name>

5. Specify the git command if you want to rename an existing remote:

git remote rename <old-name><new-name>

6. Specify the git command to fetch updates from the remote repository but not merge them into your local branch.

git fetch

7. Specify the git command to pull the latest changes from the remote repository and merge them into your local branch.

git pull

8. Specify the git command to push your local commits to a remote repository.

git push

9. Specify the git command to when pushing for the first time and want to set the remote branch you are pushing to.

git push -u origin main

10. Specify the git command to change the URL of a remote (e.g., after changing the remote repository address).

git remote set -url origin https://github.com/url

11. Specify the git command if you want to list all branches on the remote repository.

git branch -r

12. Specify the git command if a branch was deleted on the remote but still shows up locally.

git fetch -p

13. Specify the git command to fetch a specific branch from a remote.

git fetch <remotename><branchname>

14. Specify the git command if you want to see detailed information about a remote repository.

git remote show <remotename>

15. Specify the git command if you want to rebase your local branch onto a remote branch (this can be useful to keep your history linear)

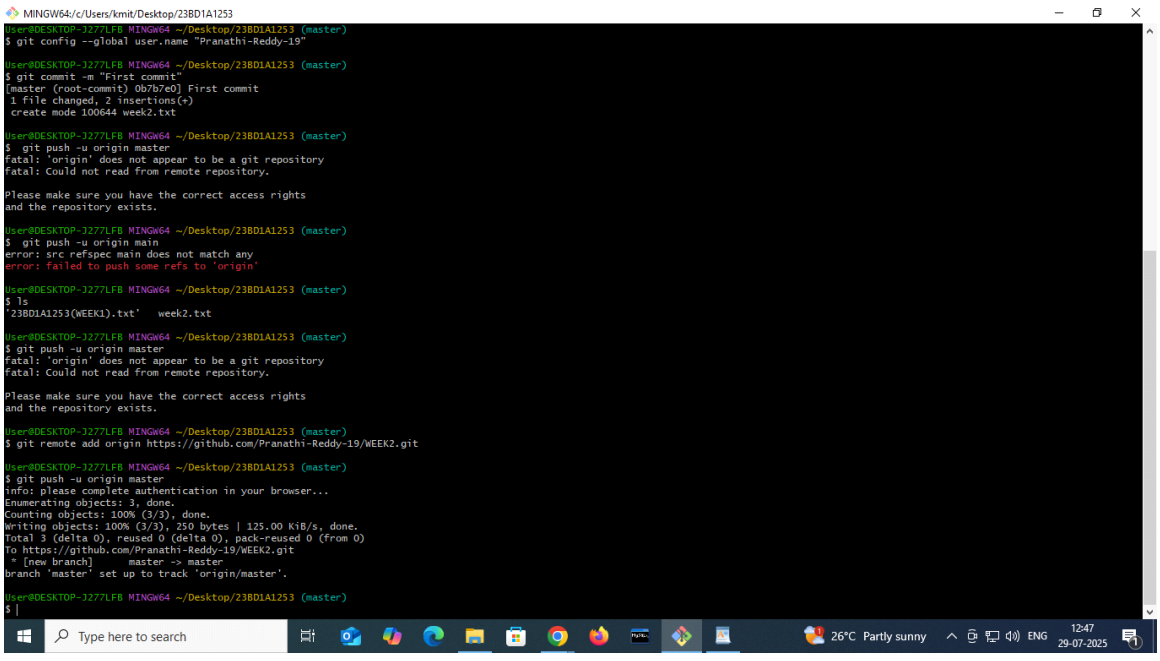
git pull --rebase origin <branchname>

**Conclusion:**

Successfully uploading a multi-folder project to both public and private GitHub repositories demonstrates student's practical understanding of **core Git commands** and **remote repository management**. By handling diverse file types (e.g., images, Java files, web pages, and pom.xml) and pushing them to different remote repositories, students show proficiency in:

- Initializing and configuring Git repositories
- Staging, committing, and managing version history
- Connecting to and working with multiple remote repositories
- Handling Gitignore and tracking large files appropriately
- Applying best practices for public vs. private repository usage

Completing these tasks equips students with essential version control skills and the ability to collaborate securely and professionally. It also reflects their readiness to contribute effectively to real-world software projects where source code management, team coordination, and secure code sharing are vital.



# **LAB** ACTION **PLAN FOR WEEK 3**

**Objective:**

1. To work on collaborative coding by:
   a. Creating Organization.
   b. Coordinating with others through a shared repository
2. To resolve conflicts when collaborating on same part of code.
3. To create and apply patch
4. Students need to make note in observation book and also upload the screenshots of above executed exercise and scenario-based question.

**1. Collaborative coding.**

Code collaboration refers to the process of multiple people working together on the same codebase.

There are **two ways** of collaborative coding:

a. Creating Organization
b. Coordinating with others through a shared repository

**a. <u>Facilitating Collaborative Work by creating Organization</u>**

<u>Organization</u> - Organizations are shared accounts where businesses and open-source projects can collaborate across many projects at once. Owners and administrators can manage member access to the organization's data and projects with sophisticated security and administrative features. Steps to facilitate collaborative work by creating organization are:

**Step 1:** Click on New organization in Git hub

Click on Create a free organization.

S**tep 2:** Set up your organization by entering the details like name, associated email, account verification, etc.

Click on Next

**Step 3**: Complete the setup by

- ✓ adding members to the group.
- ✓ Now, the invitation goes to other collaborator and they must accept.
- ✓ Next click on <u>complete setup</u>
- ✓ Goto Github and click on

**Settings → Member privileges → Base permissions → Select write → change base permission to "Write"**

Next under,
**Projects base permissions → Select Write → change base permission to "Write"**

**Step 4**: Create the remote repository for storing the project related files in Organization. This repository is accessible to every member of the team as per the permissions given.

Note: The repository can be made private so that it is accessible only to the group members rather than being in a public domain.

Now all the members of the team can contribute to the development of the project and the different files with all the versions and modification notices will be available in the respective repositories and is accessible to all the members

## b. <u>To collaborate effectively on GitHub shared repository by sharing code and coordinating with others through a shared repository.</u>

→**Steps for Collaborator 1 are**: (collaborator-1 has repository that he wants to share with collaborator-2)

- ✓ Go to Settings > collaborators
- ✓ Now in Manage Access click on Add people
- ✓ Now you will be able to see ONE collaborator added
- ✓ After this invitation goes to collaborator 2 and they need to accept it.

→**Steps for Collaborator 2 are:**

**✓ 1. Set Up Git and GitHub**

In git bash
git config --global user.name "Your Name"
git config --global user.email you@example.com

Note : run `$ git remote -v`

If
```
origin  git@github.com:OrgYOG/commonREPO.git (fetch)
origin  git@github.com:OrgYOG/commonREPO.git (push)
```
then

```
$ git remote –v // Note now, not connected to any remote
```

**✓ 2. Goto Git hub and copy url of shared repository by collaborator 1**

- Copy url (option 1)
- If you're contributing to an existing project: (option 2)
  Visit the repository page and click Fork (if you don't have write access)

**✓ 3. Clone the Repository**

Get a local copy of the repo:
In git bash
git clone https://github.com/username/repository-name.git
cd repository-name

**✓ 4. Create a Branch for Your Work**

Always work on a separate branch:
In git bash

git checkout -b feature/your-feature-name

**✅ 5. Make Changes and Commit**

Edit files, then stage and commit your changes:
In git bash to existing file / new file
git add .
git commit -m "Add a clear and descriptive commit message"

**✅ 6. Push Your Branch to GitHub**

➔ In git bash
➔ git push origin feature/your-feature-name

## Note: Following steps needed if you forked from any public repository to contribute (option 2)

**✅ 7. Open a Pull Request (PR)**

- Go to the GitHub repo in your browser
- Click Compare & pull request
- Add a meaningful title and description
- **Submit the pull request for review**

**✅ 8. Review and Discuss Changes**

- Collaborators can comment, request changes, or approve the PR(Pull Request)
- Make edits if needed and push again; they will update the PR automatically

**✅ 9. Merge the Pull Request (has to be done by both collaborators)**

Once approved:

- The repo owner can click Merge pull request
- Or you can squash and merge, depending on the team's workflow

**✅ 10. Keep Your Branch Updated (has to be done by owner collaborator 1)**

Before making new changes:
In git bash
git checkout main
git pull origin main
git checkout feature/your-new-feature
git merge main

## 2. To resolve conflicts when collaborating on same part of code.

Resolving conflicts when collaborating with GitHub (or Git in general) is a normal part of team development. Conflicts happen when two people make changes to the same part of the code, and Git doesn't know which version to keep.

**✅ Steps to Resolve a Conflict**

**1. See Which Files Are in Conflict** (Say f1.txt)
In git bash
git status
Conflicted files will be marked like:

In git bash
both modified: (Say f1.txt)

---

3. **Open the File and Look for Conflict Markers by clicking on**

   ✓ Click on **Resolve Conflict**

Git will insert conflict markers in the file like this:

```
def greet():
<<<<<<< HEAD
   print("Hello from First collaborator ")
=======
   print("Hello from second collaborator ")
>>>>>>> feature/second collaborator branch
```

This shows the two conflicting changes:

- HEAD is your version
- The section below ======= is the other branch's version or second collaborator branch version

---

**3. Edit the File to Fix the Conflict**

✓ Choose one version or combine them:

```
def greet():
    print("Hello from First collaborator ")
    print("Hello from second collaborator ")
```

✓ Then **delete** all the **conflict markers** (<<<<<<<, =======, >>>>>>>).

---

**4. Mark the Conflict as Resolved**

Once you've edited and saved the file:

✓ git add f1.txt

---

**5. Complete the Merge or Pull**

✓ git commit # Only needed if you're merging

**Or** if you're rebasing:

✓ git rebase --continue

---

⚠️ **If You Want to Abort the Merge**

If it gets messy and you want to cancel:

git merge --abort

---

Here below screen shot shows conflict raised when collaborator 1 wanted to merge changes to same line that already collaborator 2 added the line of code at that line in same file but in their own branch.

**Example :**

**Merge Conflict – Changes made to a file by two different users leads to merge conflict as below.**

**Say I am collaborator 2, now**

➔ **Accept invitation from collaborator 1**
➔ **Next goes to Shared repository by collaborator 1 (say abcrepo)**
➔ **Now open Git bash and clone the shared repo as below steps**

**Step 1: first connect to this "abcrepo" to local repository by clone**

- ✓ Now change directory to "abcrepo"
- ✓ Switch to branch feature-abc
- ✓ Now add line in README.md file
- ✓ Now git add, commit
- ✓ Now see status and push to remote branch feature-abc
- ✓ Now go to Github and refresh or click on "abcrepo", you can see README.md

**Step 2:** Now collaborator 2 adding line in README.md, save, git add, commit, push. No conflicts

**Step 3:** Also collaborator 1 adding line in README.md at same line collaborator 2 added, then git add, commit, push gives **merge conflict.**

Now open file, Edit the File to Fix the Conflict by keeping either or both collaborator changes and remove conflict markers. Add file, commit, and push.

### 3. To create and apply patch

A patch in Git is a text file that represents changes between two sets of files, or commits. It's essentially the output of the git diff command, packaged in a format that can be applied to another set of files.

Steps:

1. Goto github and take public project url
2. Clone into git bash
3. Now open file and edit it, add, commit
4. Run git log
5. Create patch with commit hash code as:

   git format-patch -1 commit-hash-code

   This creates 0001-commit-from-cloned-person.patch file.

Note: git format-patch -1 <commit-hash>

- -1 means "create a patch from 1 commit"
- This creates a .patch file for a single commit.
- Example:

  git format-patch -1 abc1234
- This creates a file like 0001-Your-commit-message.patch

To create patches for the **last 3 commits**: git format-patch -3

or

*Create patches from multiple commits*
git format-patch <base_commit>..HEAD

6. Get the path of above patch file and email or what's up to the remote owner of file.

7. Once the remote owner gets the patch file 0001-commit-from-cloned-person.patch next the owner need to apply patch file in his git bash using git command below:

Syntax:

git apply my-changes.patch

Or, if it's a patch made by git format-patch, use:

git am 0001-Your-commit-message.patch

- ✓ **git apply 0001-Your-commit-message.patch**

## 4. Scenario based questions on working with remote repository

1. You've cloned a repository and made some changes to a local branch. Now you want to push these changes to the remote repository, but you're getting an error saying "rejected - non-fast-forward." How would you resolve this?

- Answer:

Run:
git pull --rebase origin branch-name
Resolve any conflicts, then:
git push origin branch-name

2. You've been working on a feature branch, and now you need to push it to the remote repository. However, the remote repository already has a main branch. How do you push your feature branch without affecting the main branch?

- Answer:

Run:
git push origin feature-branch-name

3. You cloned a remote repository, but after a while, the repository's structure changed and new branches were added. How would you keep your local repository updated with the latest changes from the remote repository?

- Answer:

Run:
git fetch --all
git checkout new-branch-name

4. A colleague has pushed some changes to the main branch, but you have local changes in the same branch. You want to pull their changes, but you want to avoid merge conflicts. What steps would you take?

- Answer:

Run:
git add .
git commit -m "your changes"
git pull --rebase origin main

5. You accidentally pushed a sensitive file (e.g., API keys) to the remote repository. How would you fix this situation?

- Answer:

Run:
git rm --cached sensitive_file
Add to .gitignore, commit.
Use git filter-repo or BFG to clean history.
git push --force

6. You're working on a feature branch, and your manager requests that you integrate the latest changes from main into your feature branch. What steps would you take?

- Answer:

Run:
git checkout feature-branch
git pull origin main

7. You cloned a remote repository, but later you find that you need to push your changes to a different remote repository. How do you configure your local repository to push to this new remote?

- Answer:

Run:
git remote set-url origin new-url
git push -u origin branch-name

8. After running git pull, you notice that your local branch is behind the remote branch. How would you proceed to bring your local branch up to date without losing your local changes?

- Answer:

Stash or commit local changes first, then:
git pull --rebase origin branch-name

9. You're working on a project with multiple collaborators, and you notice that your local changes conflict with changes that have been pushed by others. How would you resolve the conflicts?

- Answer:

Run:
git pull --rebase origin branch-name
Resolve conflicts, then:
git add .
git rebase --continue

10. You've pushed a feature branch to a remote repository, but now you need to delete the branch from the remote. How would you do that?

- Answer:

Run:
git push origin --delete feature-branch-name.

## Scenario:

You are working on a collaborative project hosted on GitHub with a team of four developers. The main branch is main, and each developer works on their own feature branches.

You are assigned to implement a new UI component in a branch called feature/ui-update. Meanwhile, another team member has made a critical bug fix and pushed it directly to the main branch. When you try to push your changes, your push fails due to upstream changes.

Additionally, a teammate has submitted a patch file with a small CSS fix and shared it via email. You need to apply the patch, test it, and merge everything into main without breaking any functionality.

11. What command will you use to bring your local main branch up to date with the remote repository before merging it into your feature branch?

- Answer:

Run:
git checkout main
git pull origin main

12. How will you update your feature/ui-update branch to reflect the latest changes from main?

- Answer:

Run:
git checkout feature/ui-update
git merge main

13. Which command should you use to attempt pushing your local feature branch again, and what should you do if it fails due to conflicts?

- Answer:

Run:
git push origin feature/ui-update
If conflicts occur, resolve them, commit, and push again.

14. How do you apply a .patch file provided by your teammate and include it in your commit history?

- Answer:

Run:
git am 0001-Your-commit-message.patch

15. After successful testing, describe the steps (with commands) to merge your feature branch into main and push it to GitHub.

- Answer:

Run:
git checkout main
git pull origin main
git merge feature/ui-update
git push origin main

## Conclusion:

The effective use of Git in student collaborative projects promotes **teamwork, organization, and clear communication**, allowing students to manage code efficiently and track contributions accurately. By following structured Git workflows and best practices—such as regular commits, feature branching, and resolving conflicts early—students gain valuable experience in real-world development processes. This not only enhances their technical skills but also prepares them for **professional teamwork in software engineering**, where version control and collaboration are essential. Ultimately, mastering Git empowers students to deliver cleaner, more reliable code and succeed in collaborative environments.

**LAB ACTION PLAN FOR WEEK 4**

## Objective:

To enable students to:

- Understand the structure and lifecycle of a Maven project.
- Build and package Java and Web applications using Maven.
- Add dependencies using **pom.xml**, compile and test using plugins.
- Resolve errors and conflicts arising from dependency mismatches.
- Work with parent and multi-module Maven projects.
- Generate executable JARs and deployable WARs using Maven.

Students must **document observations**, include **screenshots of executions**, and answer **scenario-based questions** after completing the tasks.

- ## Understanding Maven Project Structure

Maven standardizes the project structure for both Java and web-based applications. src/

├── main/

│ ├── java/ → Application source code

│ └── resources/ → Configuration files like config.properties

└── test/

├── java/ → Unit test source code

└── resources/ → Test resources

The compiled files and reports are generated in the **`target/`** directory after a successful build.

- ## Steps to Perform Maven Build and Testing

**----mvn clean install**

- **`clean`:** Deletes the previous build (**`target/`** folder)
- **`install`:** Builds, tests, and installs the package to local **`.m2`** repository

After execution:

- **`target/`** folder contains compiled **`.class`** files and the final **`.jar`** or **`.war`**
- Artifact is stored in: **`~/.m2/repository/groupId/artifactId/version/`**

Add a Dependency (e.g., Gson):

In **`pom.xml`**:

*<dependency>*

*<groupId>com.google.code.gson</groupId>*

*<artifactId>gson</artifactId>*

*<version>2.10</version>*

*</dependency>*

After running:

*mvn clean install*

Check:

- Gson JAR is downloaded to **`.m2/repository/com/google/code/gson/gson/`**
- If version is wrong or not found → **`BUILD FAILURE`** with dependency resolution error

- **Configure Java Version via Compiler Plugin**

In **`pom.xml`**:

*<plugin>*
*<groupId>org.apache.maven.plugins</groupId>*
*<artifactId>maven-compiler-plugin</artifactId>*
*<version>3.11.0</version>*
*<configuration>*
*<source>17</source>*
*<target>17</target>*
*</configuration> </plugin>*

Check:

- Run `mvn package`
- Inspect generated JAR: *jar tf target/myapp.jar*

- **JUnit Testing and Reports**

Place test files in `src/test/java`. Example:

```
public class AppTest {
   @Test
   public void testSum() {
      assertEquals(5, 2 + 3);
   }
}
```

Run:

mvn test

Check:

- `target/test-classes/` → compiled test `.class` files
- `target/surefire-reports/` → `.txt` or `.xml` test results
  If test fails → corresponding log and failure trace shown

- **Handling Errors in pom.xml**
  - Typos in version numbers or missing repositories cause `BUILD FAILURE`
  - Maven shows precise error in console
  - Fix the dependency tag → re-run `mvn clean install`

- **Adding Resource Files**

  Put config.properties inside:

  src/main/resources/config.properties

  To read:

  ```
  InputStream input =
  getClass().getClassLoader().getResourceAsStream("config.properties");
  ```

  After build, check target/classes/ → file should exist there.

- **Multi-Module and Parent Projects**

  Structure:

*parent/*
*├── pom.xml (packaging: pom)*
*├── core/*
*│ └── pom.xml*
*└── web/*
└── pom.xml

  - Parent `pom.xml` defines `<modules>` and common dependencies
  - Each submodule builds independently into its `target/` directory

- **Executable JARs**

  Add to `pom.xml`:

```
<build>
 <plugins>
  <plugin>
   <groupId>org.apache.maven.plugins</groupId>
   <artifactId>maven-jar-plugin</artifactId>
   <configuration>
    <archive>
     <manifest>
      <mainClass>com.example.Main</mainClass>
     </manifest>
    </archive>
   </configuration>
  </plugin>
 </plugins>
</build>
```

  Run:

```
mvn package
java -jar target/myapp.jar
```

- **Building a WAR File**

  Create a Maven web project with structure:

  ```
  src/main/webapp/
   └── WEB-INF/web.xml
  ```

  Add:

  *<packaging>war</packaging>*

  *Command:*

  *mvn package*

  Generates target/mywebapp.war → deploy on Tomcat server.

- **Scenario-Based Questions:**

  - If my error is about:

    *Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.8.1:compile*

    *[ERROR] -> [Help 1]*

    *[ERROR] To see the full stack trace of the errors, re-run Maven with the -X switch.*

    How can I resolve it using maven terminal?
  - A dependency you added is not recognized by the compiler. What steps would you take to confirm it is available in **`.m2`** and listed in dependency tree?
  - A teammate sends a **`.patch`** file for a bug fix. How would you apply it and include it in your Maven build?
  - You have multiple **JUnit** test failures and want to rerun only failed tests. How would you approach this?
  - Your Maven build fails due to "Unsupported class version error." What plugin and configuration would you review?
  - You need to change your Java application from a **`WAR`** to a standalone **`JAR`**. What `pom.xml` changes are needed?
  - You are required to change the default build output directory from **`target/`** to **`build_output/`**. How would you configure it?
  - You want to skip tests during the Maven build. What command would you use?
  - How would you generate a site report (with test coverage, dependency analysis) for a Maven project?

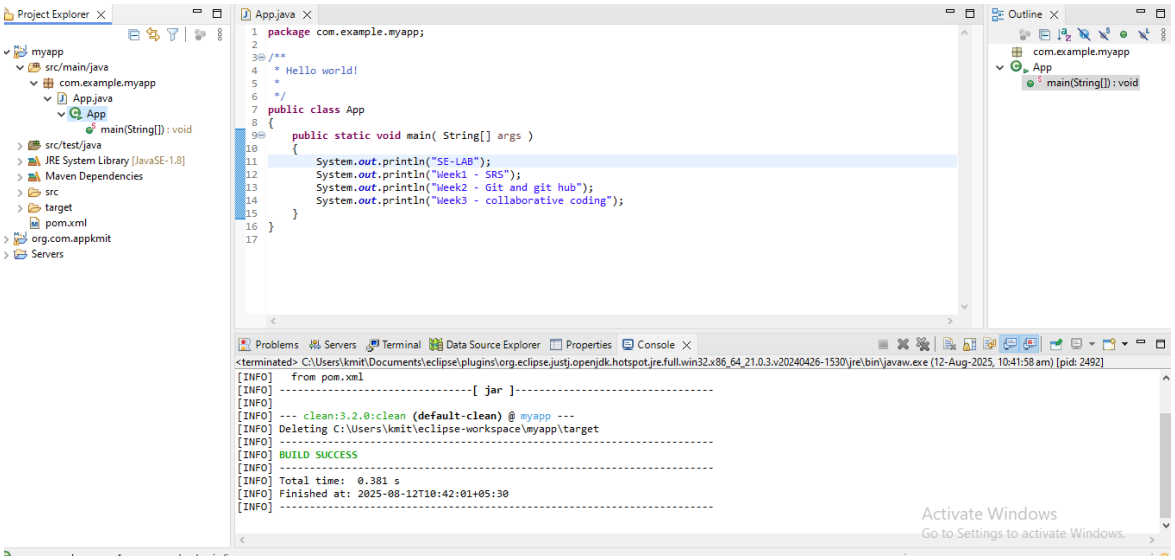
- How do you build a Java project using Maven, and what files are generated in the **target/** folder after running mvn clean install?
- How does Maven resolve dependency conflicts when two libraries use different versions of the same dependency, and how can you view and manage the **dependency tree**?
- How do you write and run a JUnit test in a Maven project, and where are the compiled test classes and reports stored after running **mvn test**?
- How can you create an executable **JAR** with a main method using Maven, and which **plugin** helps configure this behavior?
- How do you install and use a custom third-party JAR file in your Maven project, and how can you confirm it's included in the build and **classpath**?
- How do you create a Maven web project that packages into a WAR file, and what is the standard folder structure for such a project?
- What command do you use to build a WAR file in Maven, where is it generated, and how can you deploy it to a server like **Apache Tomcat**?
- How do you add JSTL and **servlet-api dependencies** in a Maven web project, and why should the servlet API use provided scope instead of compile?
- How do you set up a multi-module Maven web project with separate modules for core logic and web interface, and how are these modules built and connected?
- How do you configure a Maven web project, and how does its packaging and execution differ from a traditional WAR-based application?

## Conclusion

Mastering Maven empowers students to structure projects efficiently, manage complex dependencies, and automate testing and packaging. By practicing real-world scenarios—such as resolving build errors, handling resource files, applying patches, and deploying to servers—students gain valuable hands-on experience aligned with professional software development workflows. Maven not only streamlines builds but also enforces standardization and reusability across projects. Through Maven, students learn efficient project management, dependency control, multi-module integration, and reproducible builds. Understanding Maven's lifecycle, plugin system, and error handling prepares students for professional DevOps and CI/CD workflows. This lab equips students with hands-on experience in packaging, testing, resolving conflicts, and deploying real-world Java and Web applications.

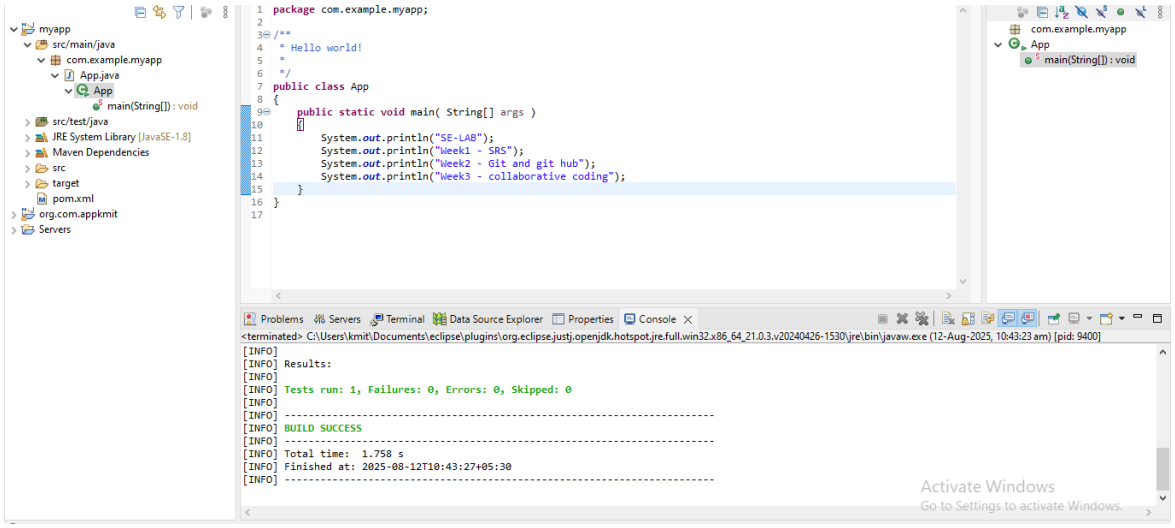
**Maven Project**

**1. maven clean**

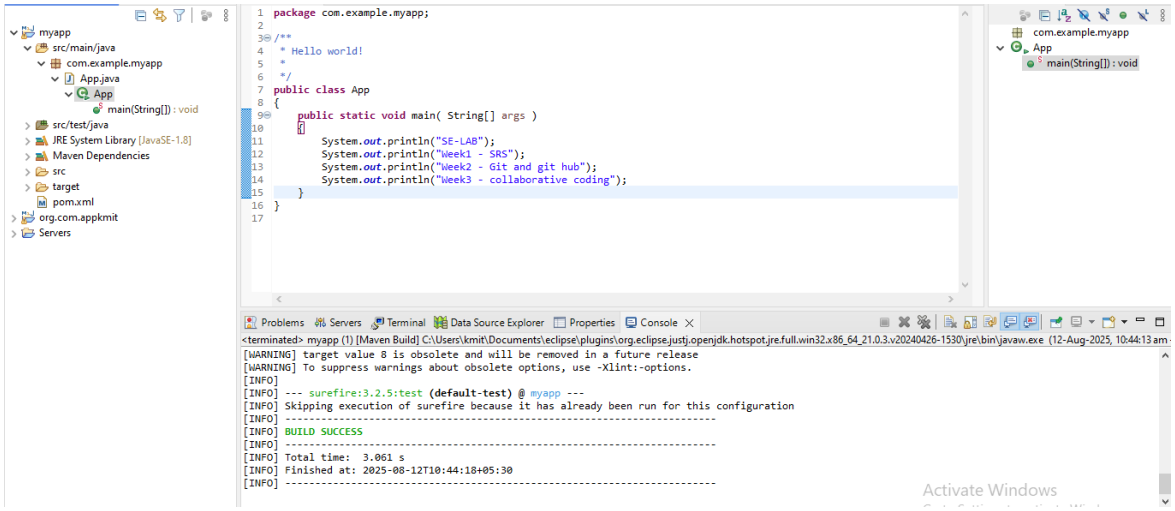



## 2. maven install

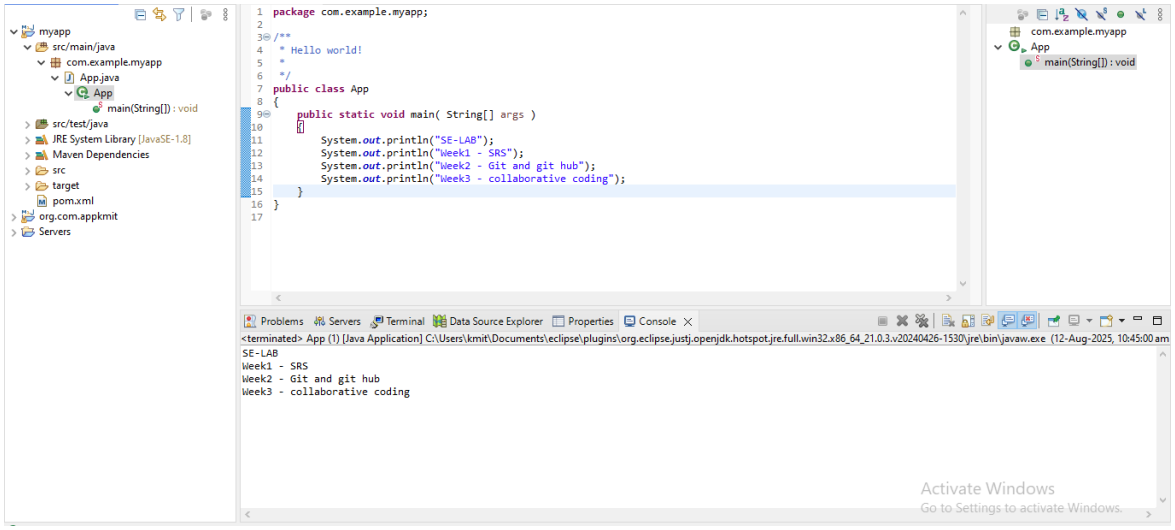

## 3. maven test



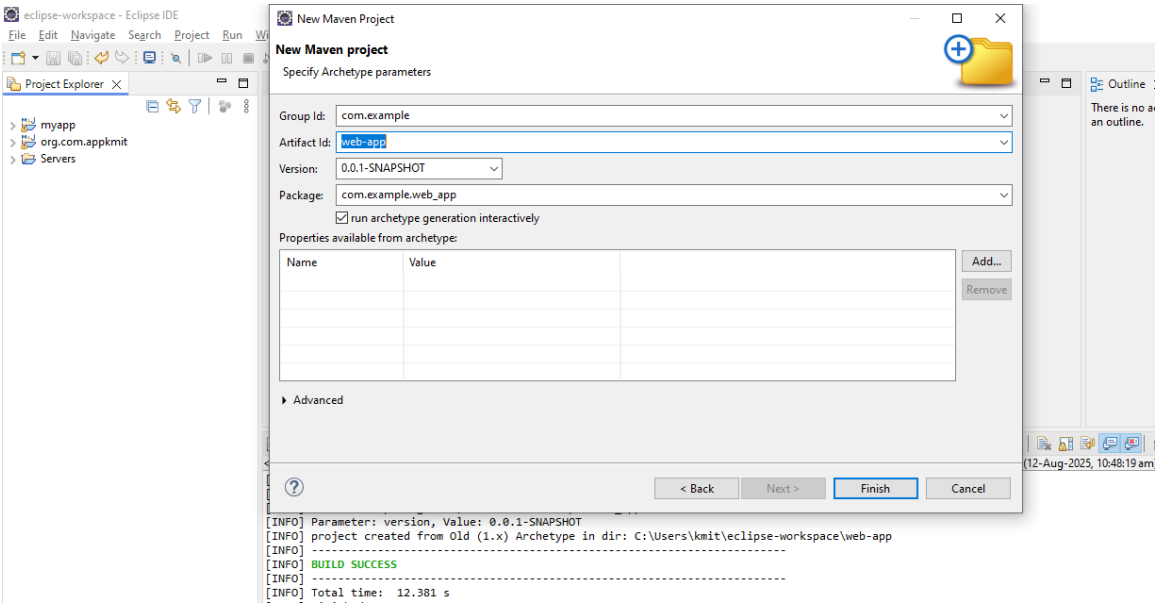

## 4. maven build

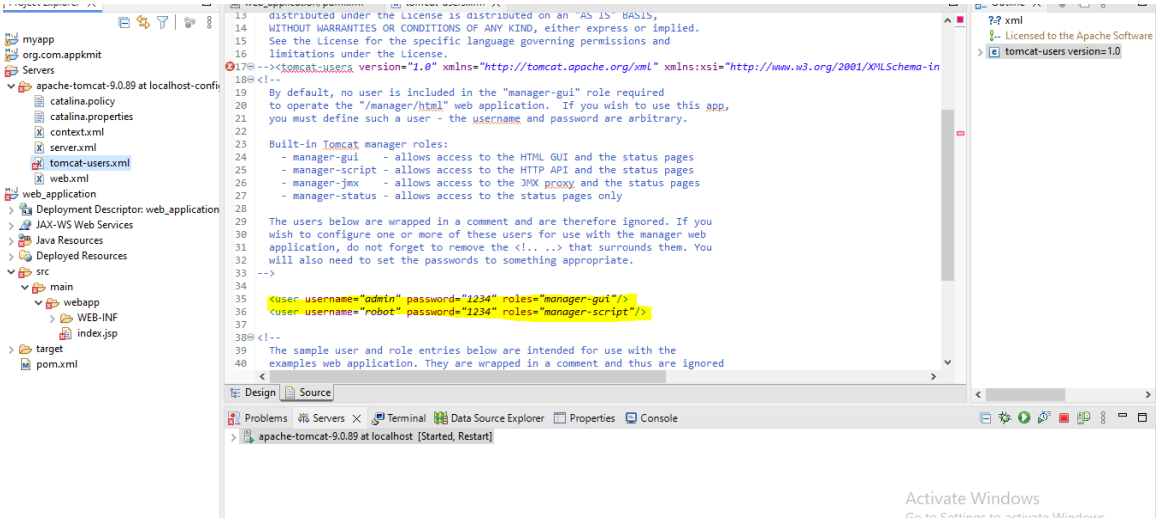


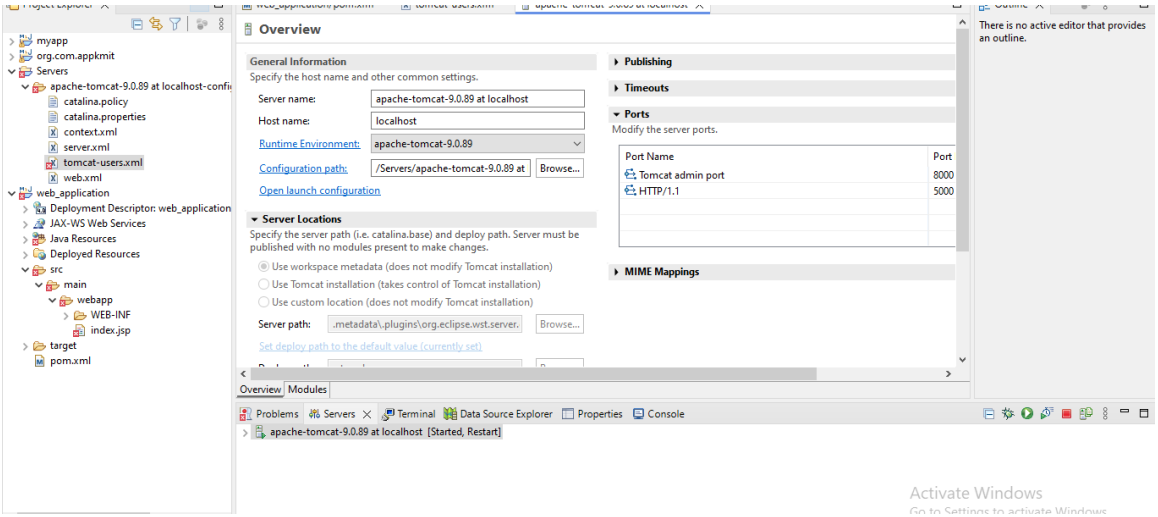

## 5. run java application

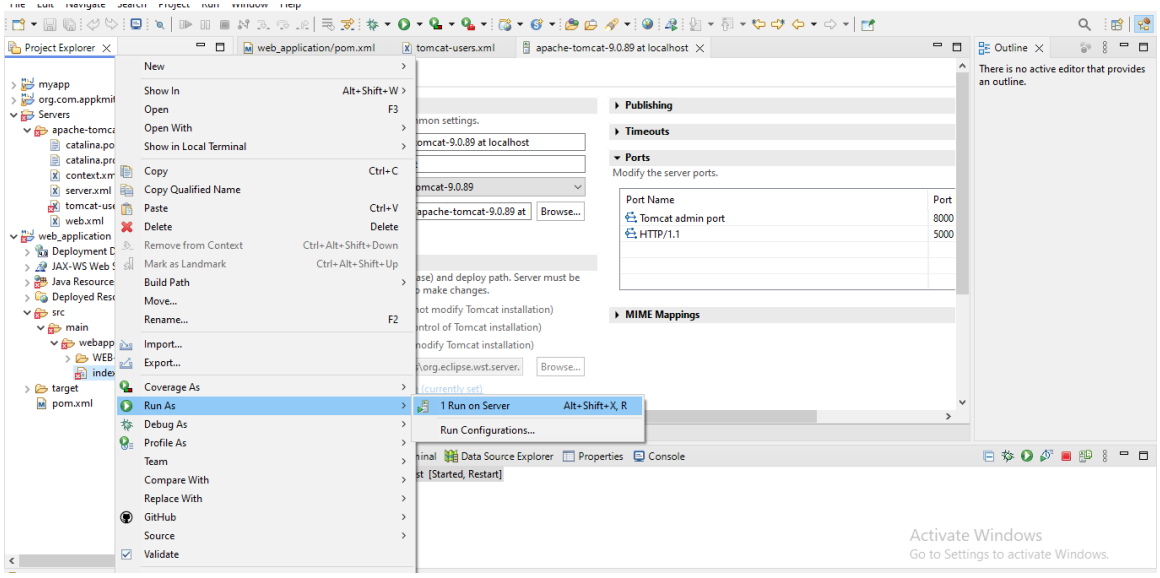


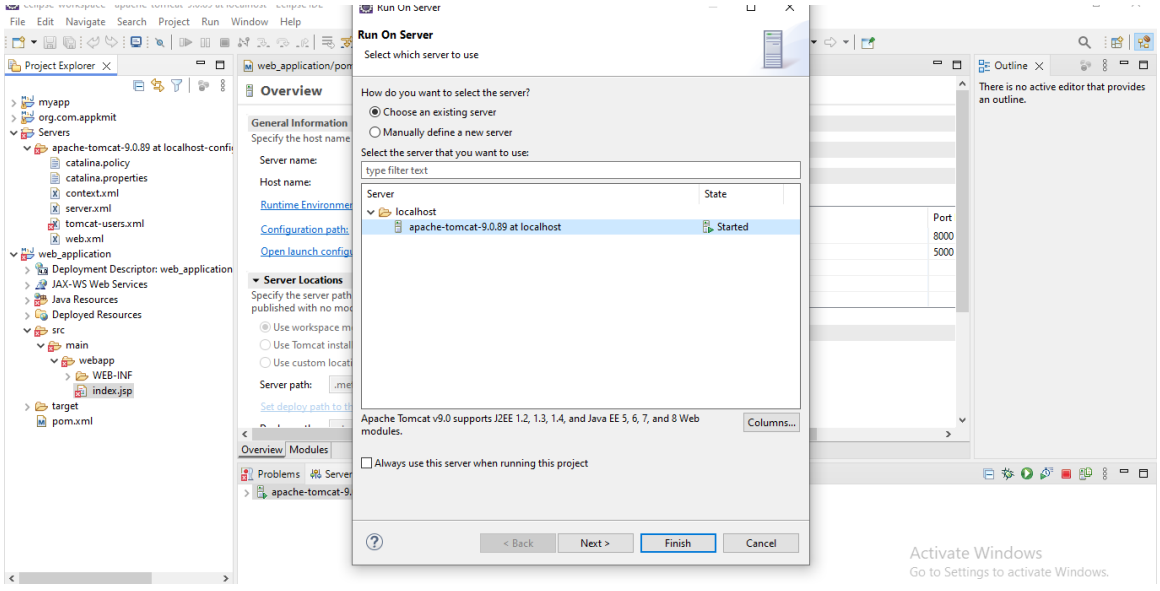

## WEB-APP

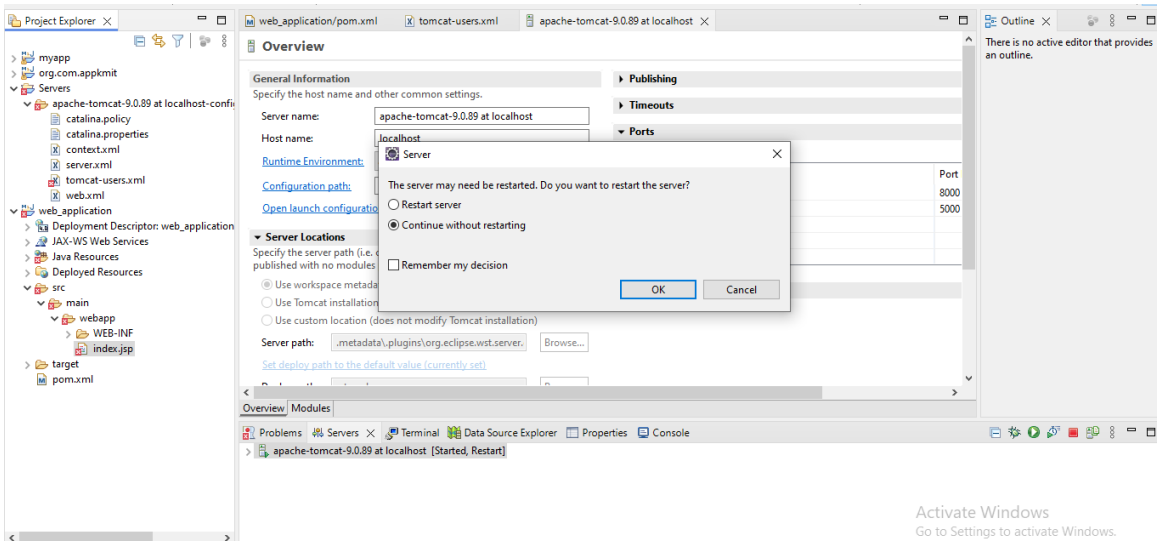

eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run
Project Explorer
myapp
org.com.appkmit
Servers
New Maven Project
New Maven project
Specify Archetype parameters
Group Id: com.example
Artifact Id: web-app
Version: 0.0.1-SNAPSHOT
Package: com.example.web_app
run archetype generation interactively
Properties available from archetype:
Name
Value
Add...
Remove
Advanced
< Back
Next >
Finish
Cancel
Outline
There is no active editor that provides an outline.
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\kmit\eclipse-workspace\web-app
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 12.381 s

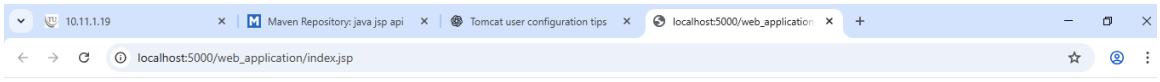

myapp
org.com.appkmit
Servers
apache-tomcat-9.0.89 at localhost-confi
catalina.policy
catalina.properties
context.xml
server.xml
tomcat-users.xml
web.xml
web_application
Deployment Descriptor: web_application
JAX-WS Web Services
Java Resources
Deployed Resources
src
main
webapp
WEB-INF
index.jsp
target
pom.xml
<user username="admin" password="1234" roles="manager-gui"/>
<user username="robot" password="1234" roles="manager-script"/>
Design
Source
Problems
Servers
Terminal
Data Source Explorer
Properties
Console
apache-tomcat-9.0.89 at localhost [Started, Restart]
Activate Windows


Overview
General Information
Specify the host name and other common settings.
Server name: apache-tomcat-9.0.89 at localhost
Host name: localhost
Runtime Environment: apache-tomcat-9.0.89
Configuration path: /Servers/apache-tomcat-9.0.89 at
Browse...
Open launch configuration
Server Locations
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.
Use workspace metadata (does not modify Tomcat installation)
Use Tomcat installation (takes control of Tomcat installation)
Use custom location (does not modify Tomcat installation)
Server path: .metadata\.plugins\org.eclipse.wst.server.
Set deploy path to the default value (currently set)
Publishing
Timeouts
Ports
Modify the server ports.
Port Name
Port
Tomcat admin port
8000
HTTP/1.1
5000
MIME Mappings
Overview
Modules
There is no active editor that provides an outline.
Activate Windows


Project Explorer
web_application/pom.xml
tomcat-users.xml
apache-tomcat-9.0.89 at localhost
Outline
There is no active editor that provides an outline.
New
Show In Alt+Shift+W
Open F3
Open With
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Mark as Landmark Ctrl+Alt+Shift+Up
Build Path
Move...
Rename... F2
Import...
Export...
Coverage As
Run As
Debug As
Profile As
Team
Compare With
Replace With
GitHub
Source
Validate
1 Run on Server Alt+Shift+X, R
Run Configurations...
Publishing
Timeouts
Ports
Modify the server ports.
Port Name
Port
Tomcat admin port 8000
HTTP/1.1 5000
MIME Mappings
Data Source Explorer
Properties
Console
Activate Windows
Go to Settings to activate Windows.


Run On Server
Run On Server
Select which server to use
How do you want to select the server?
Choose an existing server
Manually define a new server
Select the server that you want to use:
type filter text
Server
State
localhost
apache-tomcat-9.0.89 at localhost
Started
Apache Tomcat v9.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, and 8 Web modules.
Columns...
Always use this server when running this project
< Back
Next >
Finish
Cancel
Overview
General Information
Server Locations
Overview
Modules
Activate Windows
Go to Settings to activate Windows.


Server
The server may need be restarted. Do you want to restart the server?
Restart server
Continue without restarting
Remember my decision
OK
Cancel
Overview
General Information
Server Locations
Problems
Servers
Terminal
Data Source Explorer
Properties
Console
apache-tomcat-9.0.89 at localhost [Started, Restart]
Activate Windows
Go to Settings to activate Windows.




**Hello World!**

# LAB ACTION PLAN FOR WEEK 5

## Objectives:

Students will able to:

- Learn how to pull, run, stop, start, remove, and inspect containers and images.
- Gain the ability to create, monitor, and troubleshoot running containers.
- Configure and manage networks for container communication.
- Create and manage persistent storage for containers.
- Learn how to list, remove, and manage images efficiently.

### Docker:

Docker is an open-source platform that automates the process of building, packaging, and running applications inside lightweight, portable containers, ensuring they run consistently across different environments.

### Docker Images:

A read-only template that contains the application code, runtime, libraries, and dependencies. It is like a blueprint for creating containers.

### Docker Containers:

A running instance of a Docker image.

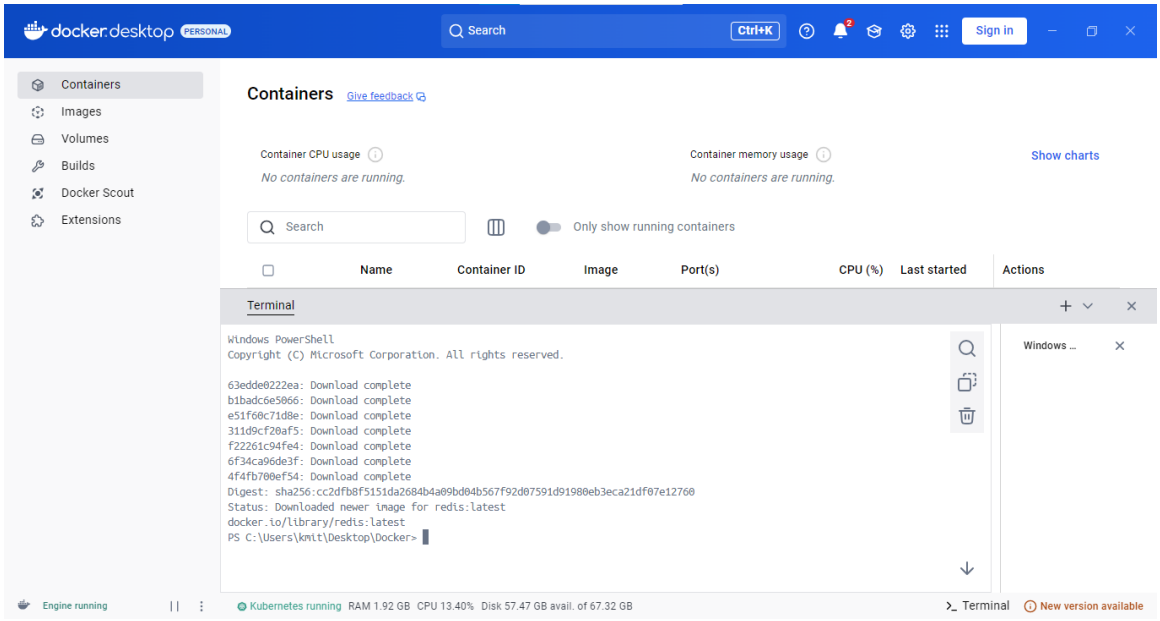
### Working with Docker CLI Commands:

### Docker CLI Commands with redis

**Step 1:** Pull the redis Image

Command:

docker pull redis




**Step 2:** Run a Redis Container

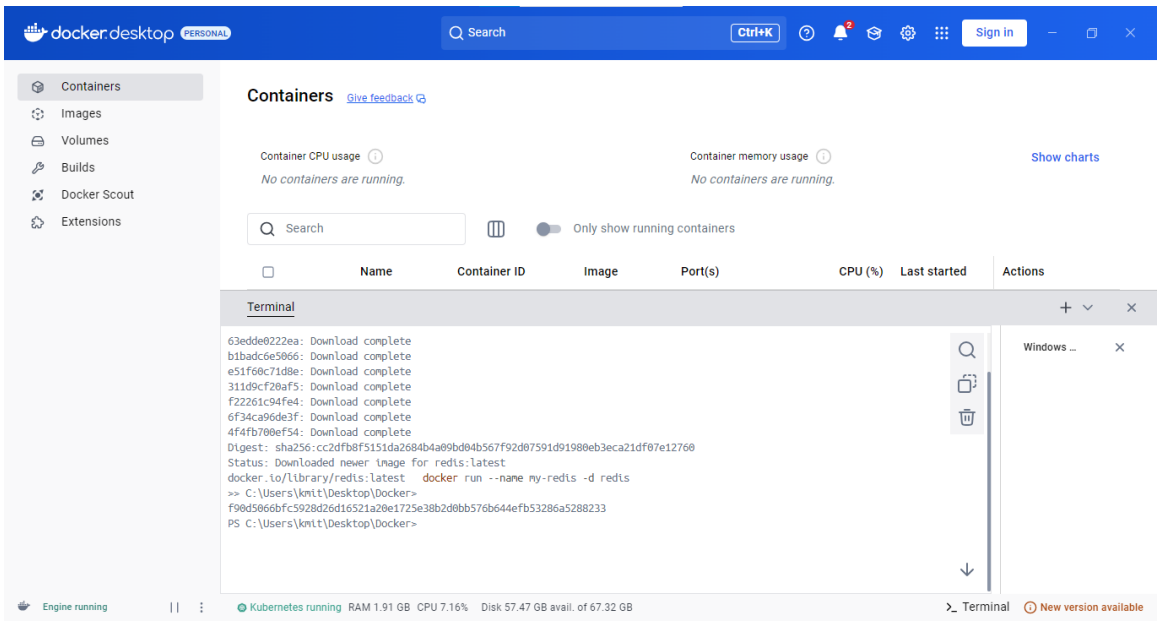
Command:

docker run --name my-redis -d redis

What It Does:

 Creates and starts a container named my-redis from the redis image.

 The -d flag runs the container in the background.



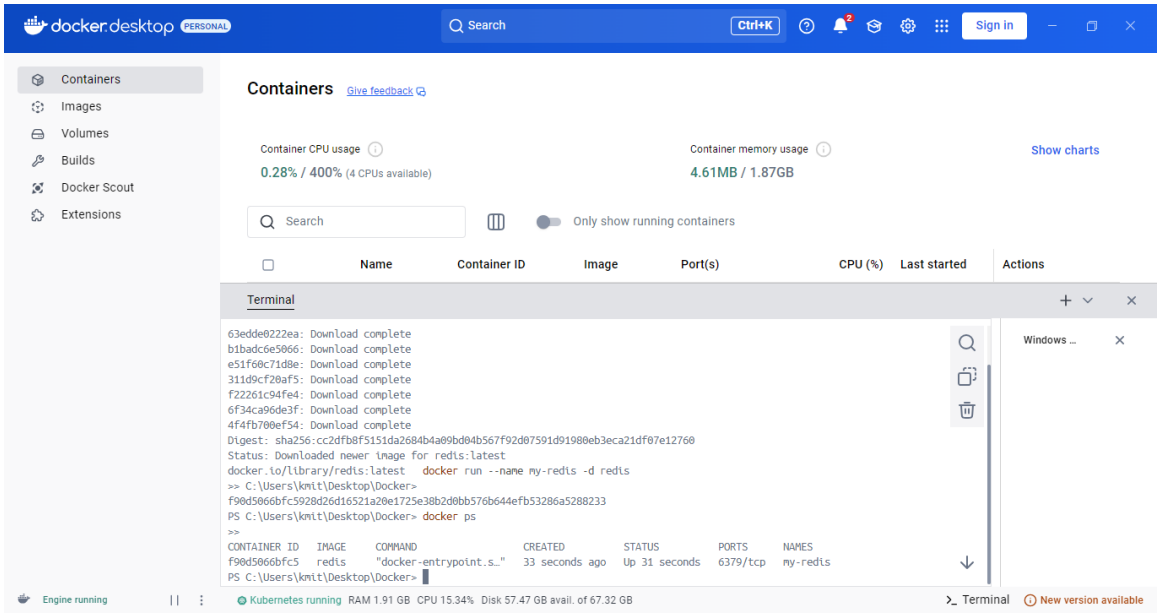

**Step 3:** Check Running Containers

Command:

docker ps

What It Does:

 Lists all running containers.




**Step 4:** Access Redis

Command:

docker exec -it my-redis redis-cli

Opens the Redis command-line tool (redis-cli) inside the container.

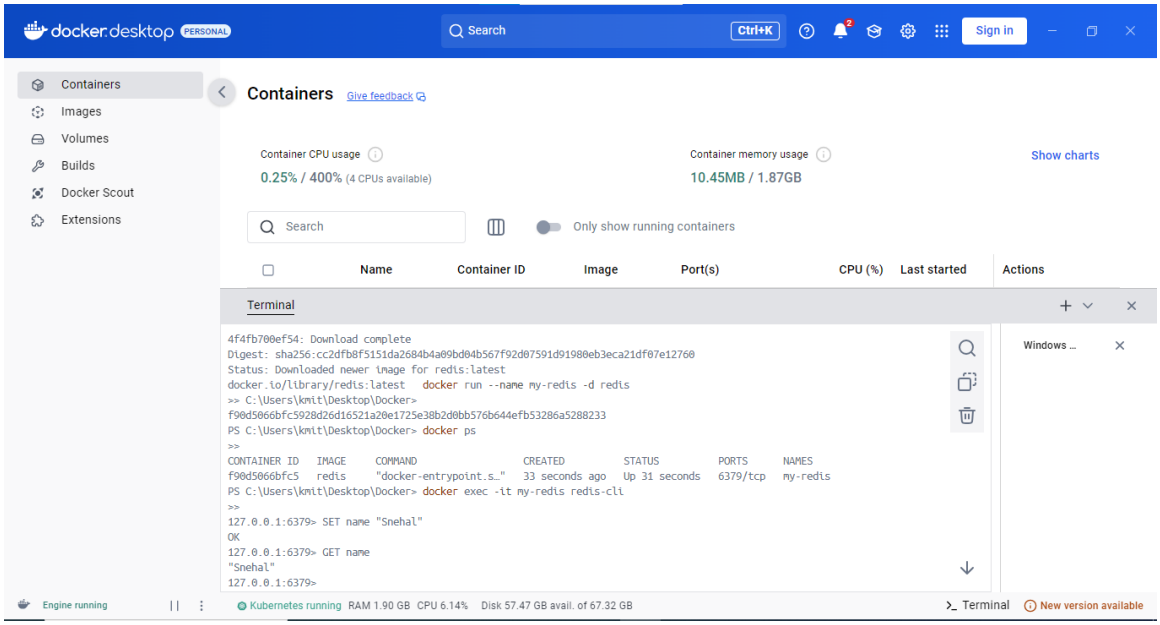
**Example** Redis Commands:

127.0.0.1:6379> SET name "Alice"

OK

127.0.0.1:6379> GET name

"Alice"



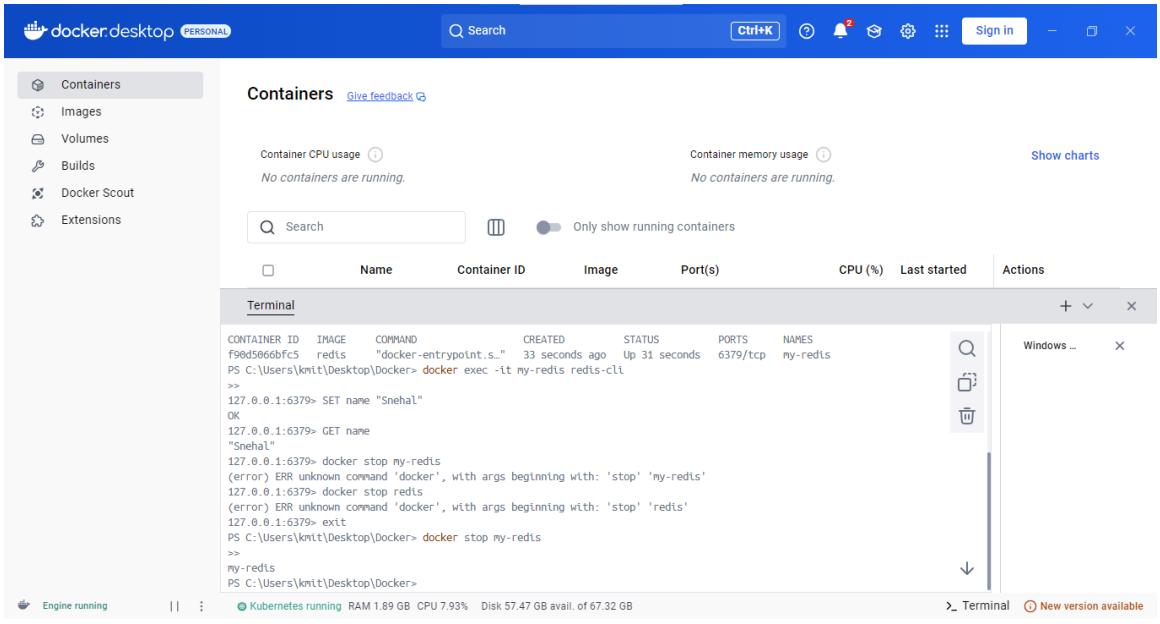

**Step 5:** Stop the Redis Container

Command:

docker stop my-redis

What It Does:

 Stops the Redis container but doesn’t delete it.



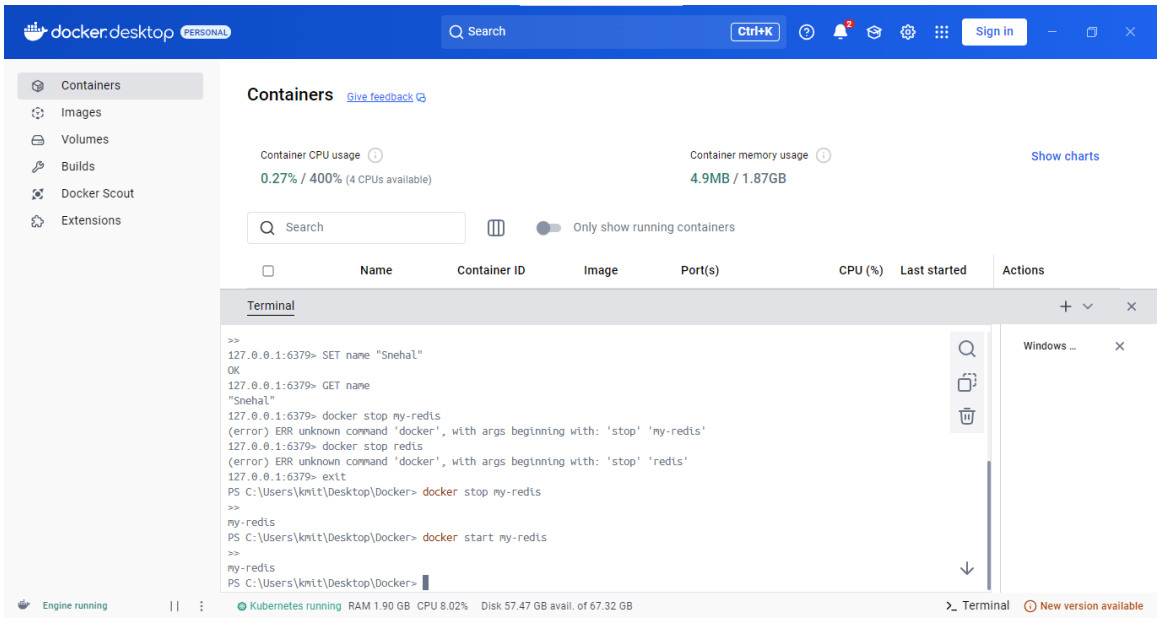

**Step 6:** Restart the Redis Container

Command:

docker start my-redis

What It Does:

 Restarts the stopped container.



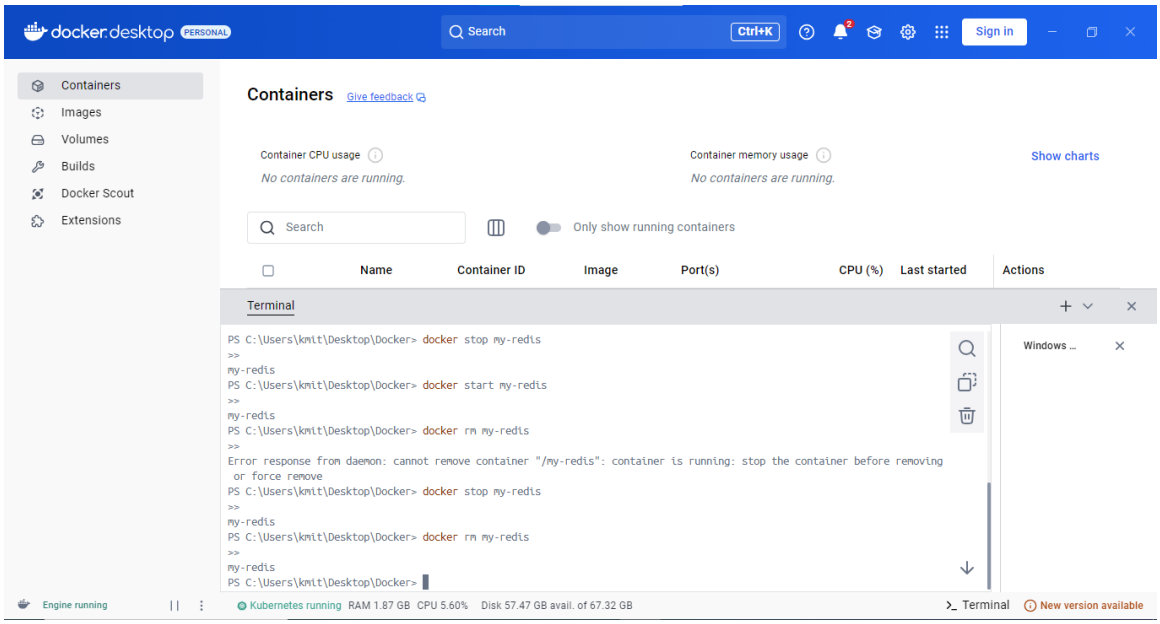

**Step 7:** Remove the Redis Container

Command:

docker rm my-redis

What It Does:

 Deletes the container permanently.



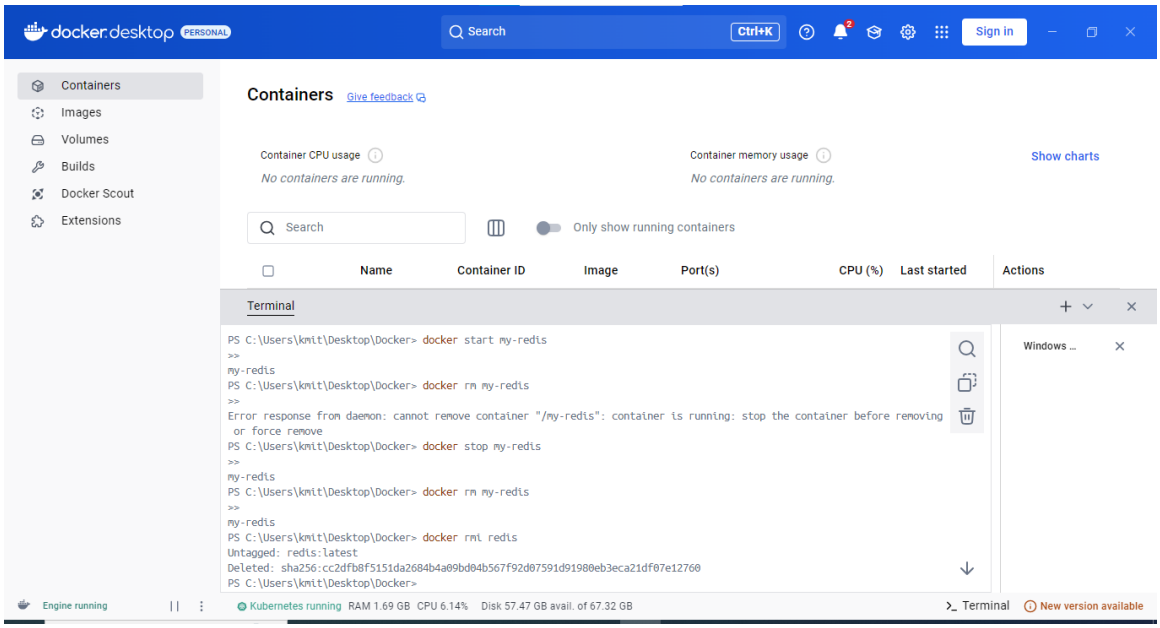
**Step 8:** Remove the Redis Image

Command:

docker rmi redis

What It Does:

 Deletes the Redis image from your local system.




## 2.Working with Docker file

A Docker file is a text file with instructions to create a custom Docker image.

**Step 1:** Set Up Your Folder

1. Windows:

o Create a folder like C:\DockerProjects\Redis.

o Open Git Bash and navigate to the folder:

cd /c/DockerProjects/Redis

2. Mac/Linux:

o Create a folder:

mkdir ~/DockerProjects/Redis

cd ~/DockerProjects/Redis

**Step 2:** Write the Dockerfile

1. Inside the folder, create a file named Dockerfile (no extension).

2. Add the following content:

**FROM redis:latest**

**CMD ["redis-server"]**

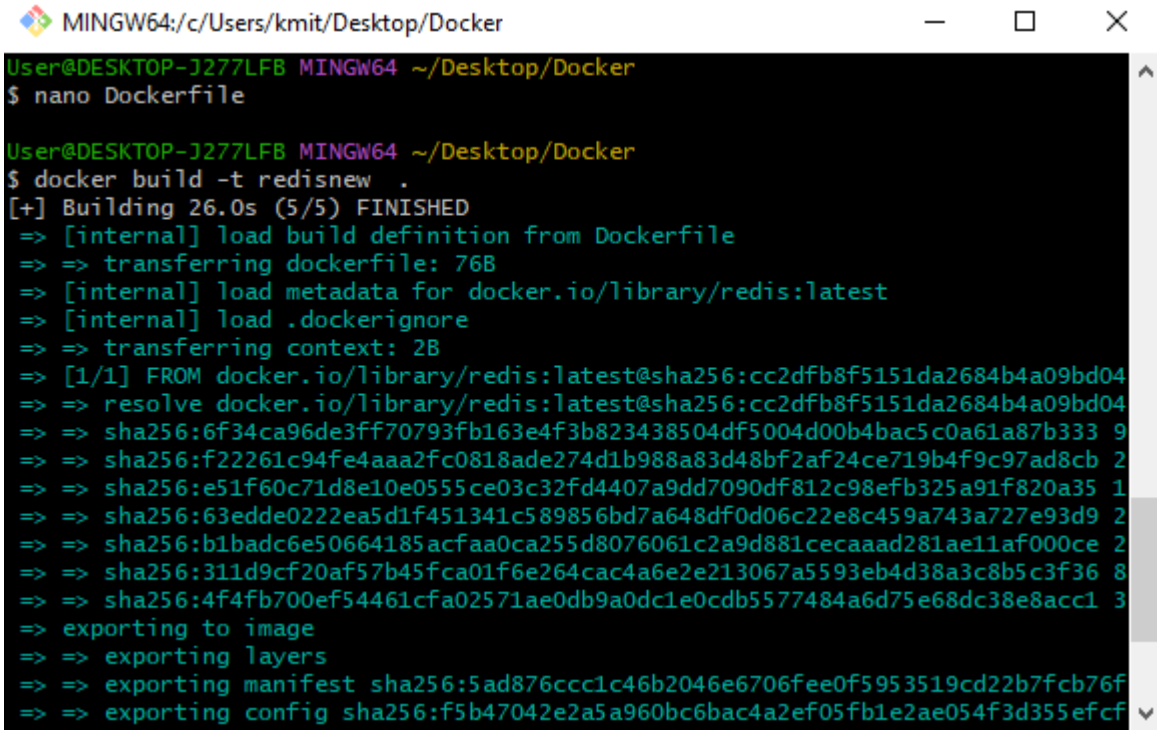
Docker Commands (Step-by-step):

**1. docker build -t redisnew .**

What it does:

 This creates (builds) a Docker image using the recipe (Dockerfile) in the current folder (.).

 -t redisnew: Gives the image a name/tag ("redisnew"), so you can find it easily.



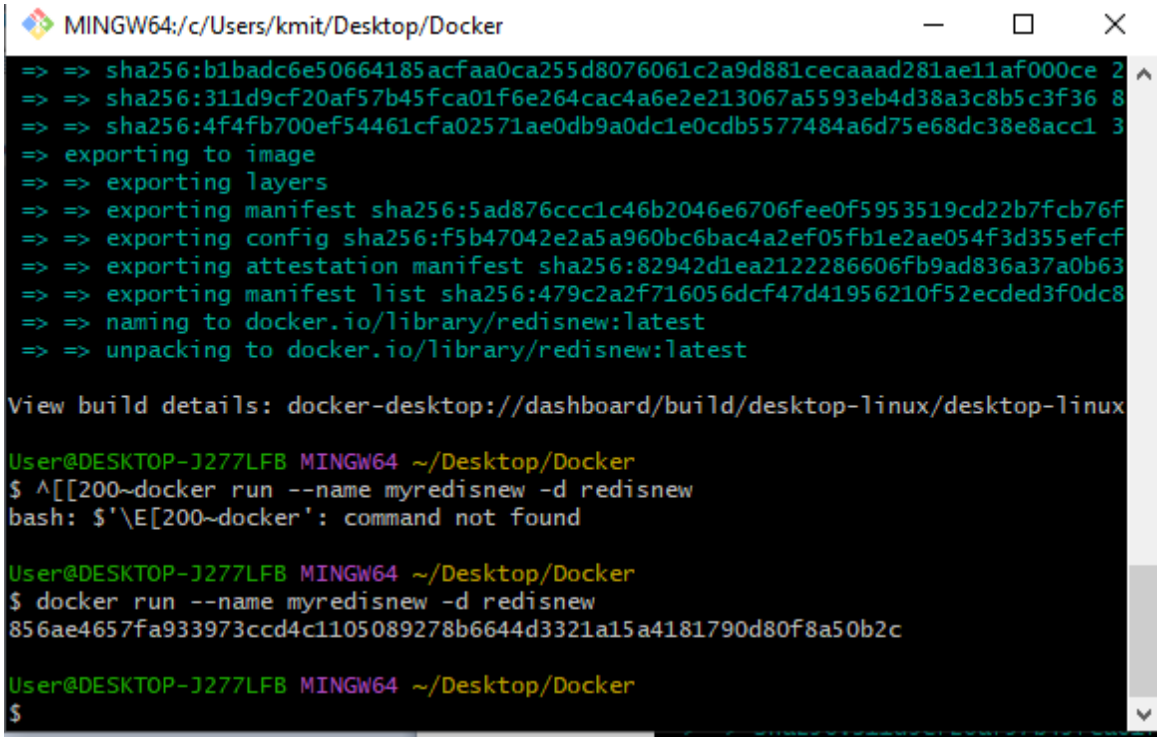

**2. docker run --name myredisnew -d redisnew**

What it does:

 Starts a new container (mini computer) from the redisnew image.

 --name myredisnew: Names the container "myredisnew" so it’s easy to identify.

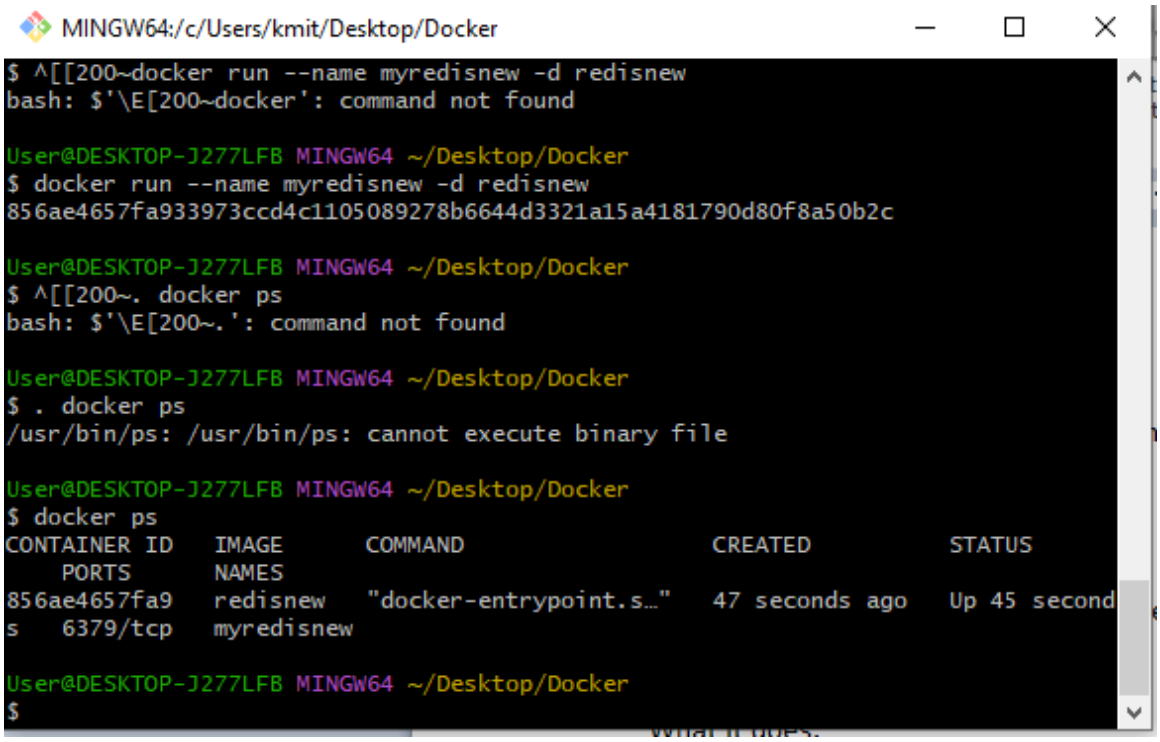
 -d: Runs the container in the background.




**3. docker ps**

What it does:

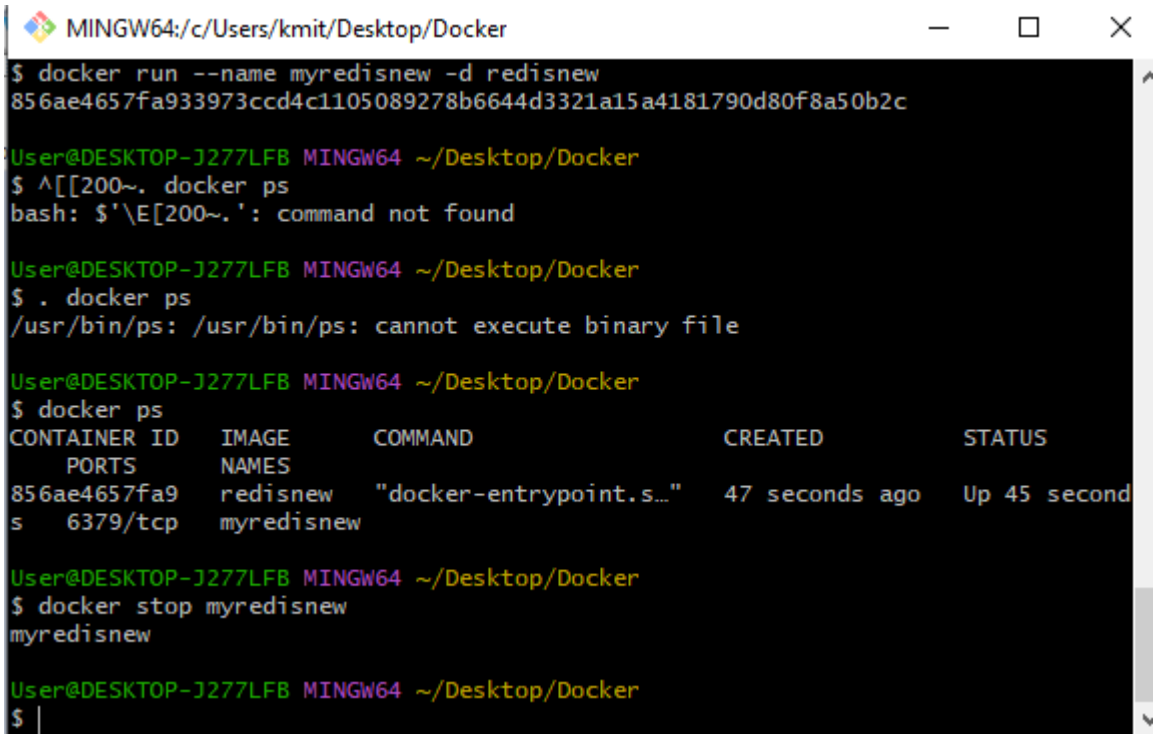
 Shows a list of containers that are running right now.




**4. docker stop myredisnew**

What it does:

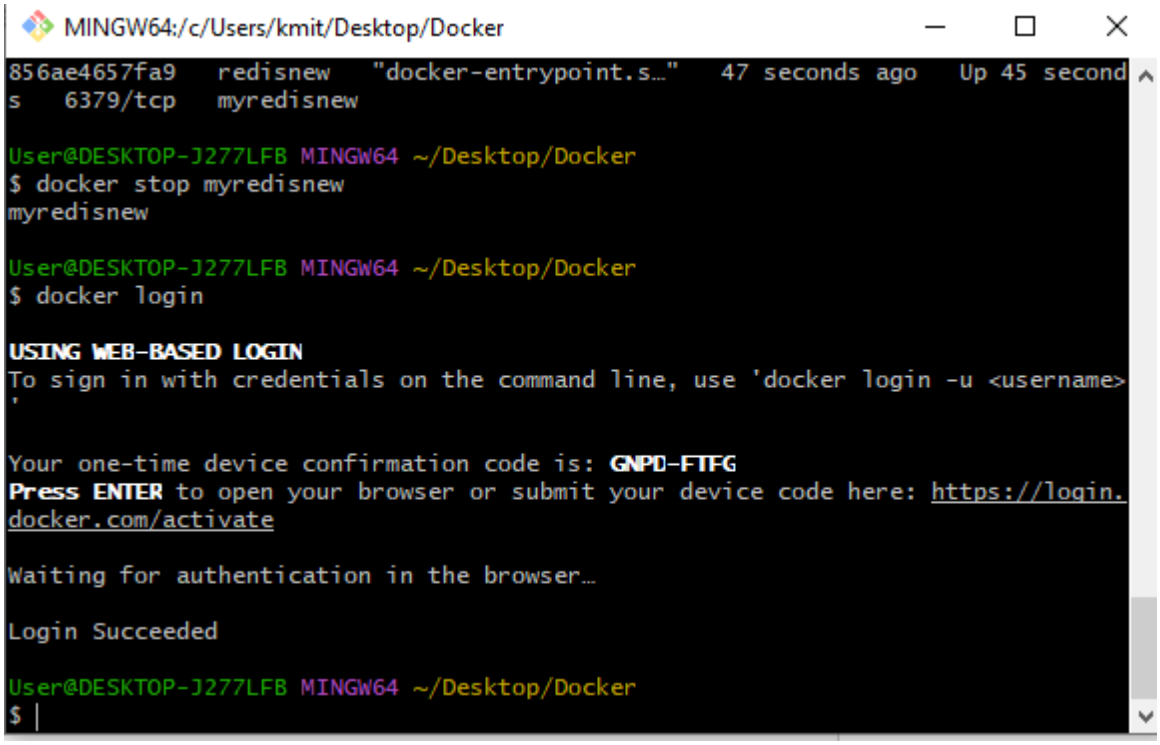
 Stops the container named "myredisnew" (like turning off a computer).




**5. docker login**

What it does:

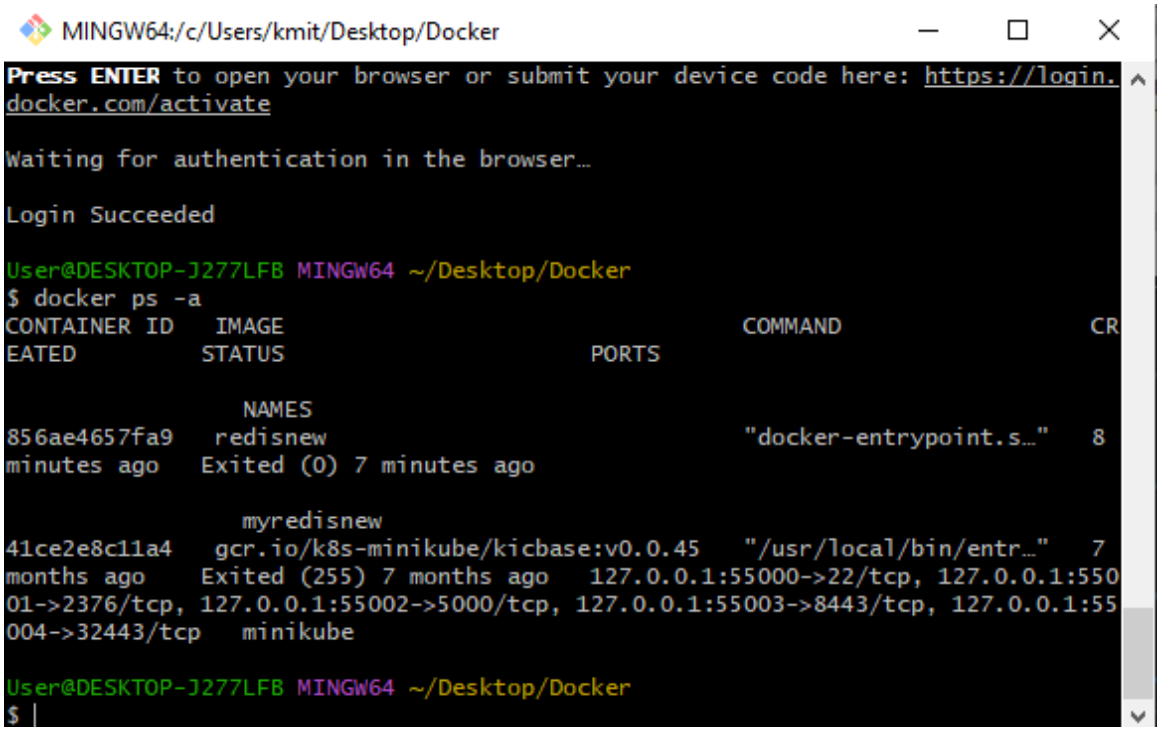
 Logs you into your Docker Hub account, so you can upload images.




**6. docker ps -a**

What it does:

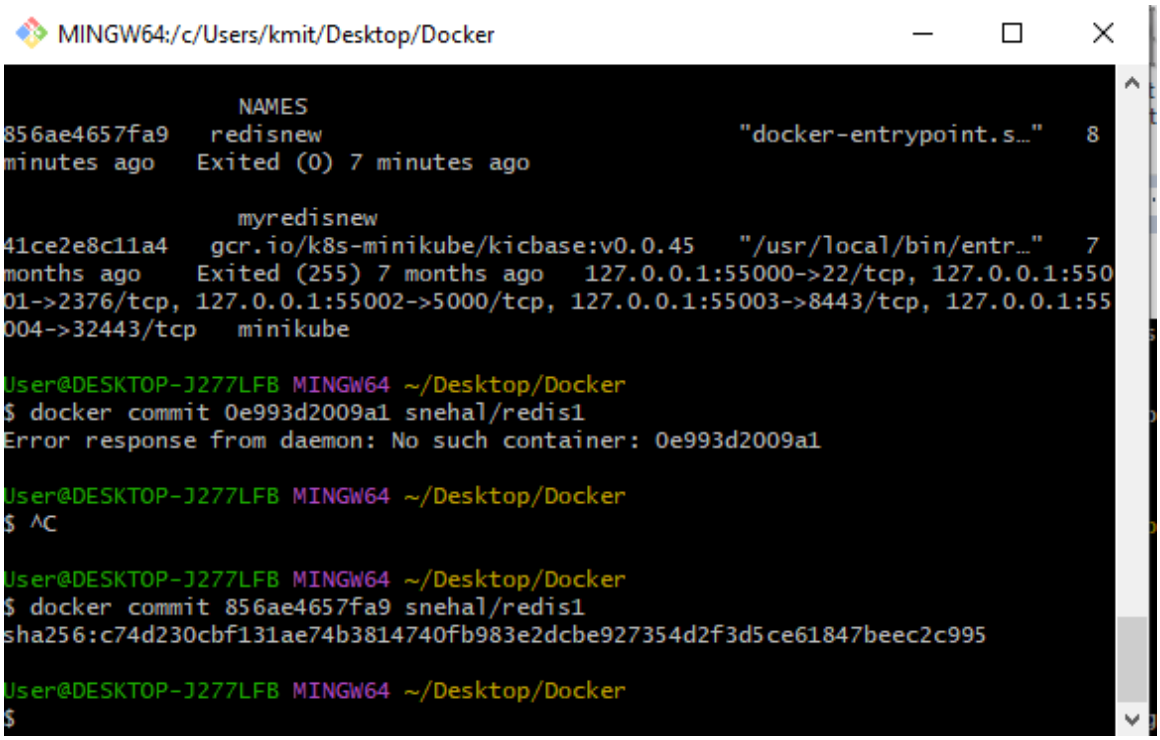
 Shows a list of all containers, including stopped ones.




**7. docker commit** `856ae4657fa9` **snehal1729/redis1**

What it does:

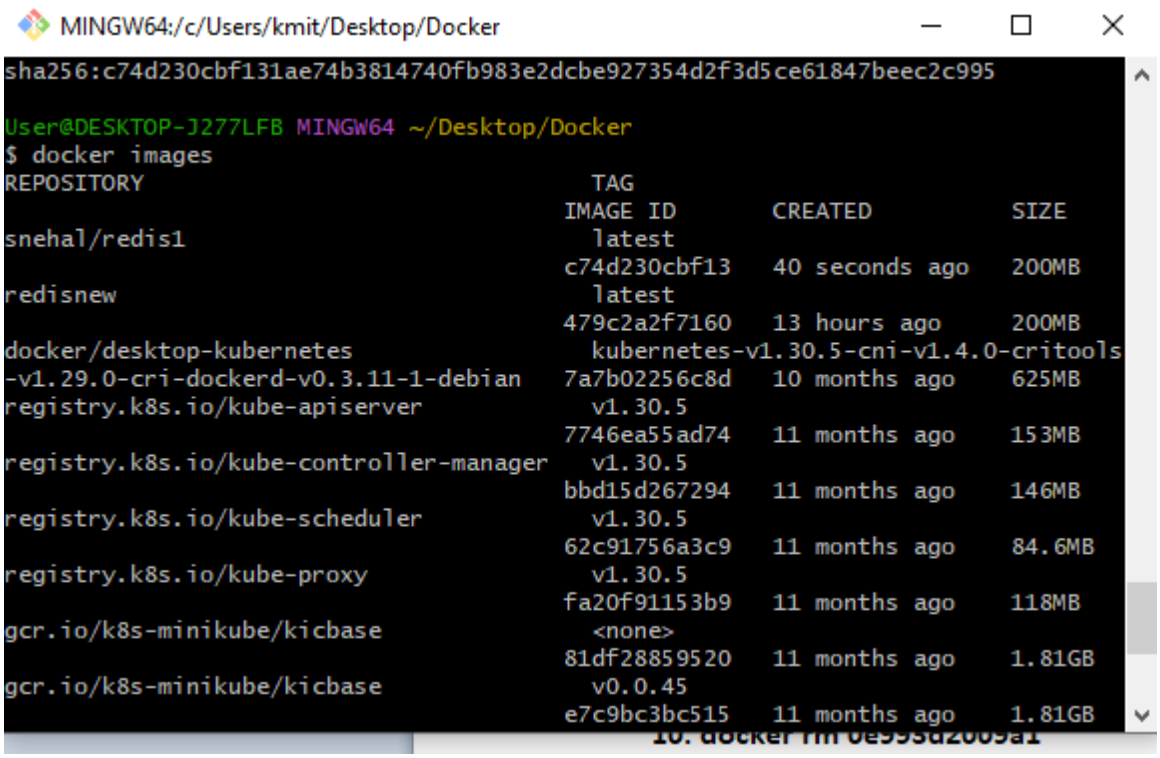
 Takes a snapshot (saves changes) of the container with ID 0e993d2009a1 and creates a new image called snehal1729/redis1.




**8. docker images**

What it does:

 Lists all images saved on your system.

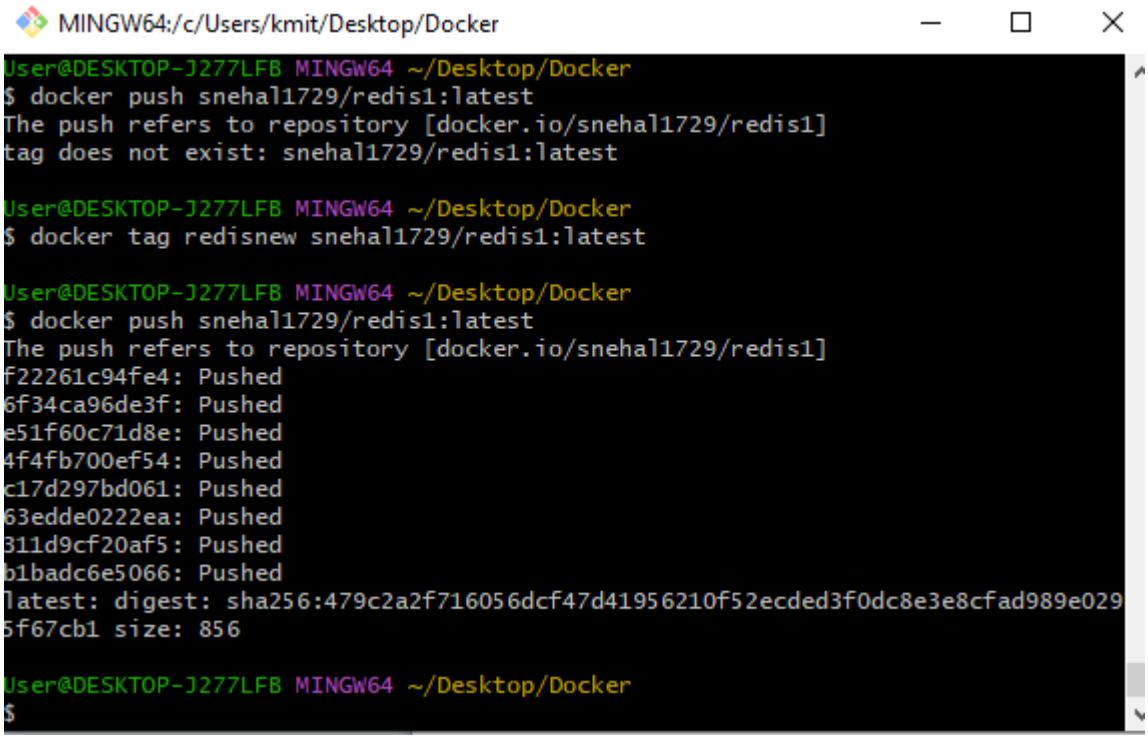



**9. docker push snehal1729/redis1**

What it does:

 Uploads the image 1729/redis1 to Docker Hub, so others can download

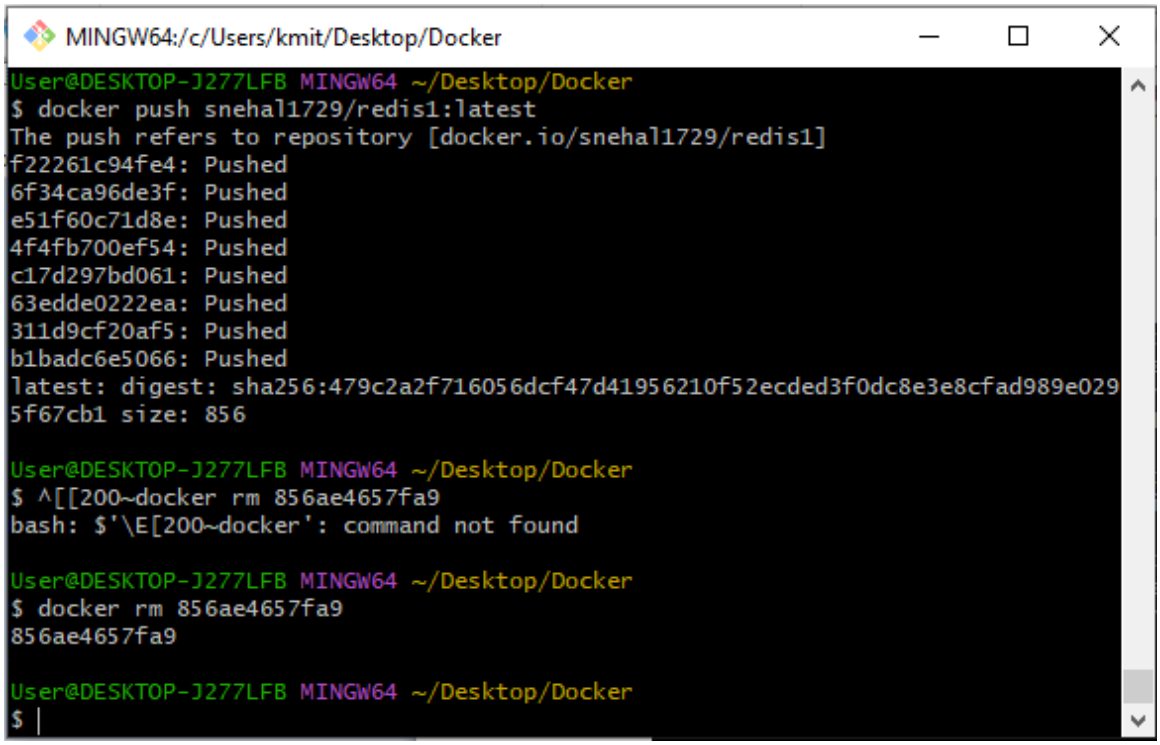
it.



**10. docker rm** `856ae4657fa9`

What it does:

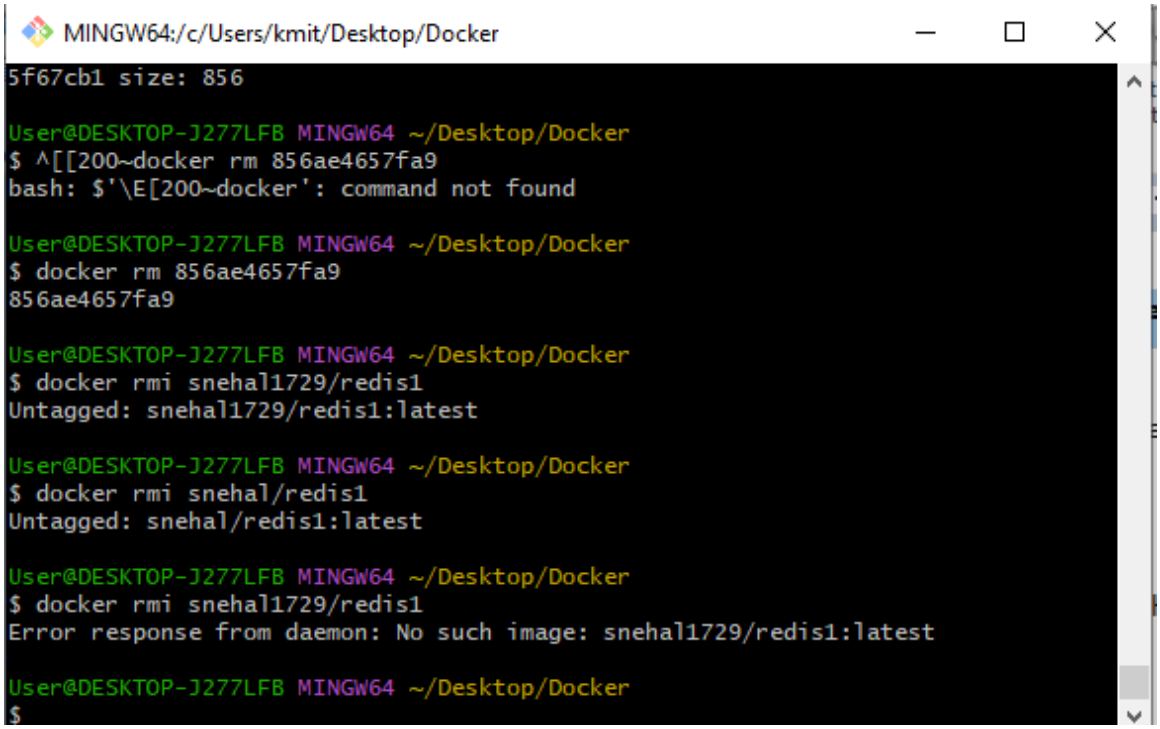
 Deletes the container with ID 0e993d2009a1.




**11. docker rmi budarajumadhurika/redis1**

What it does:

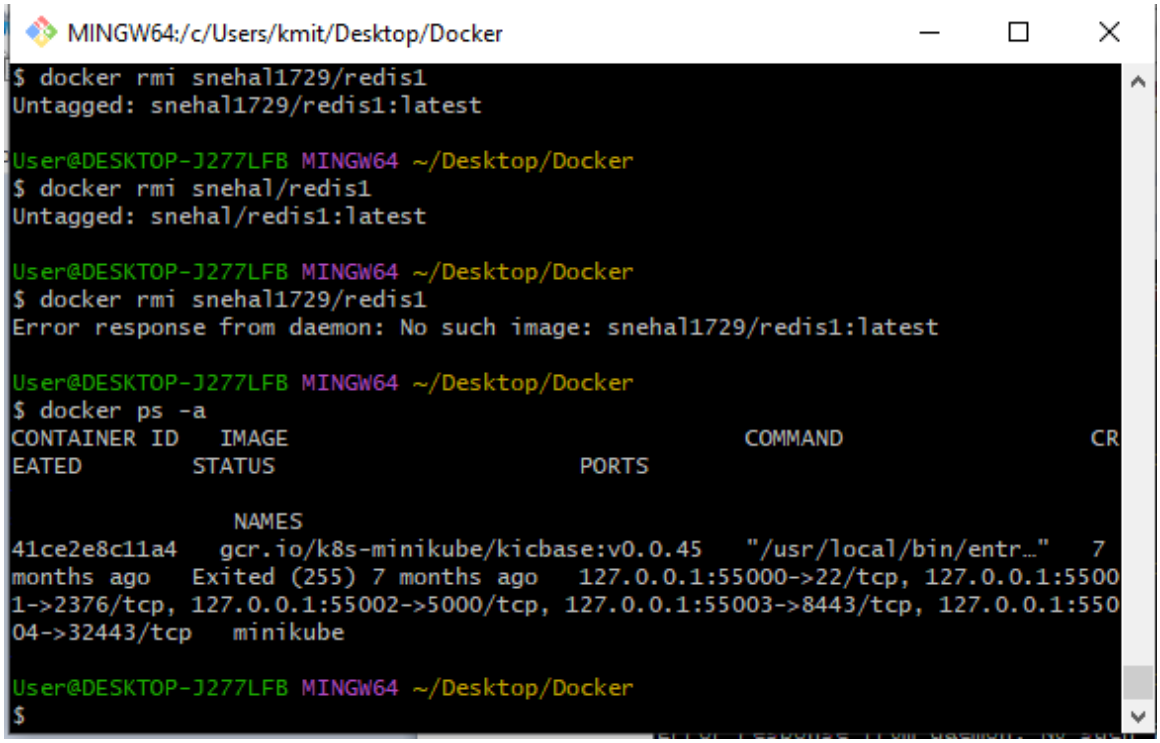
 Deletes the image budarajumadhurika/redis1 from your system.




**12. docker ps -a**

What it does:

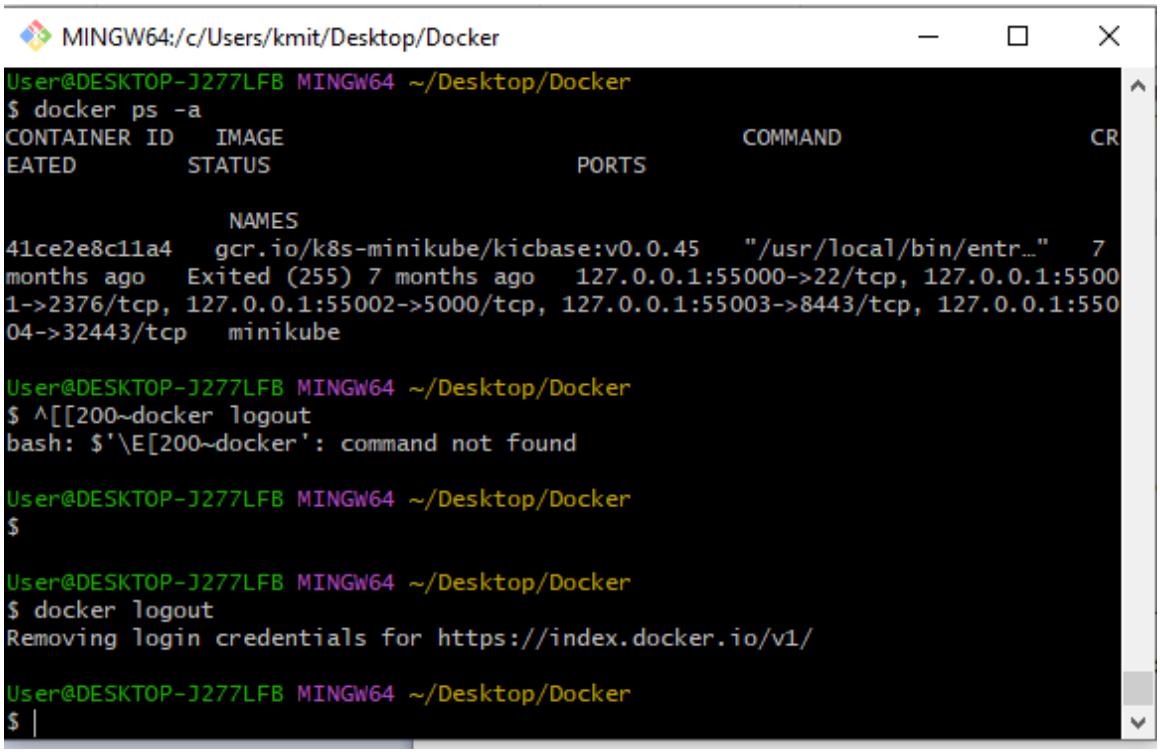
Shows all containers again to confirm changes.




**13. docker logout**

What it does:

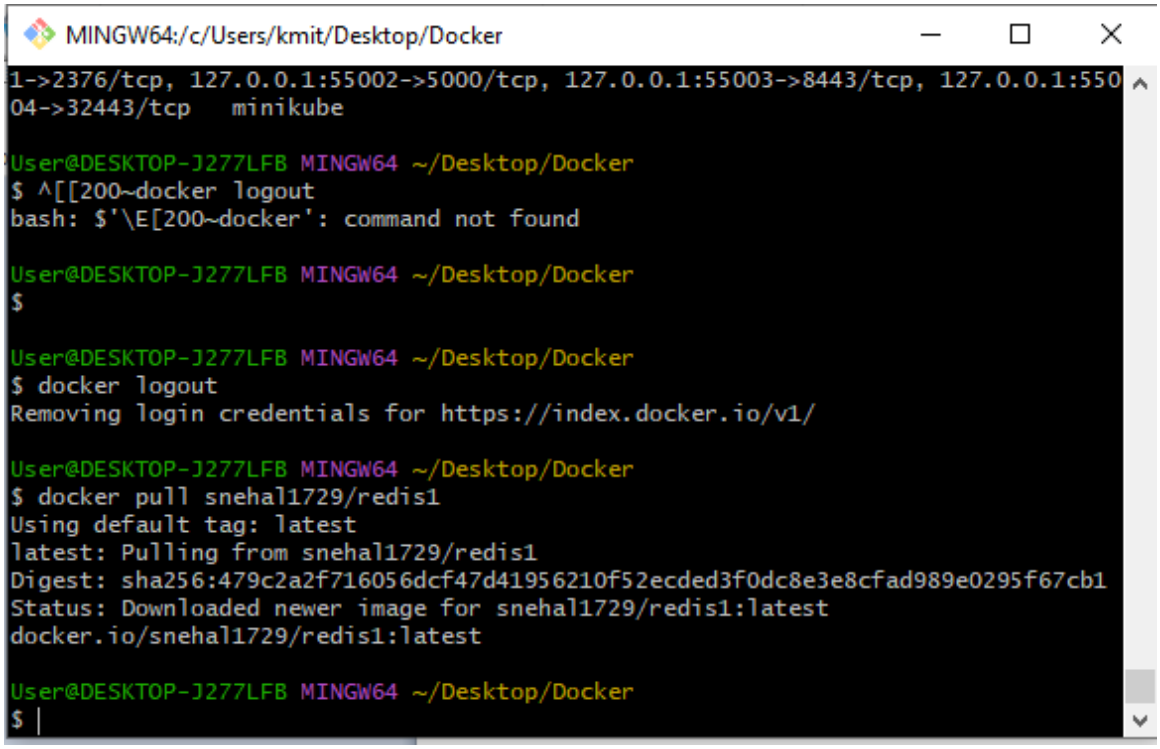
 Logs you out of Docker Hub.




**14. docker pull snehal1729/redis1**

What it does:

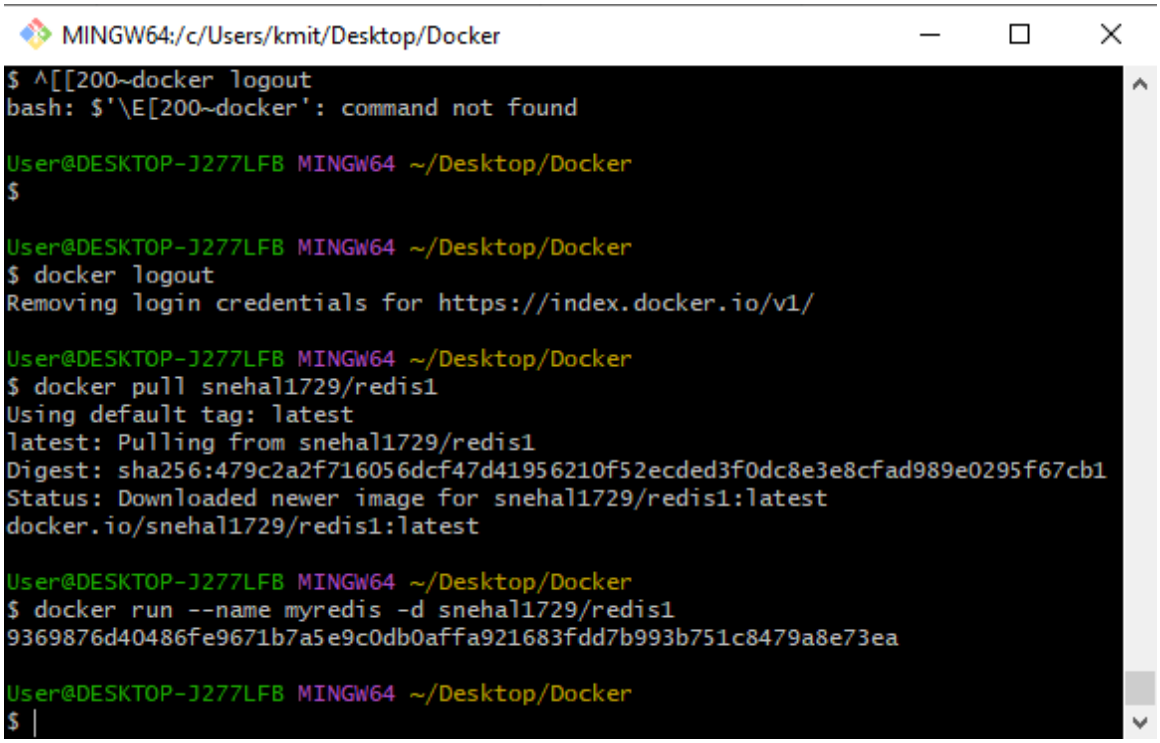
 Downloads the image budarajumadhurika/redis1 from Docker Hub.




**15. docker run --name myredis -d snehal1729/redis1**

What it does:

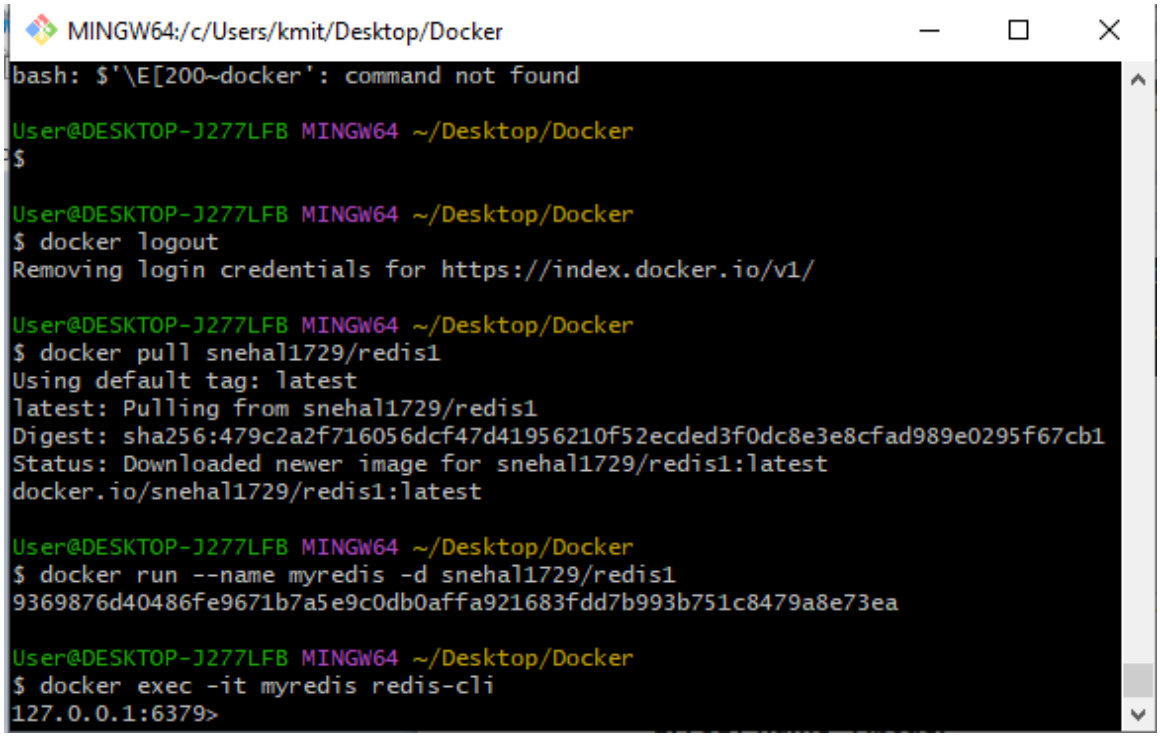
 Starts a new container using the image budarajumadhurika/redis1.



**16. docker exec -it myredis redis-cli**

What it does:

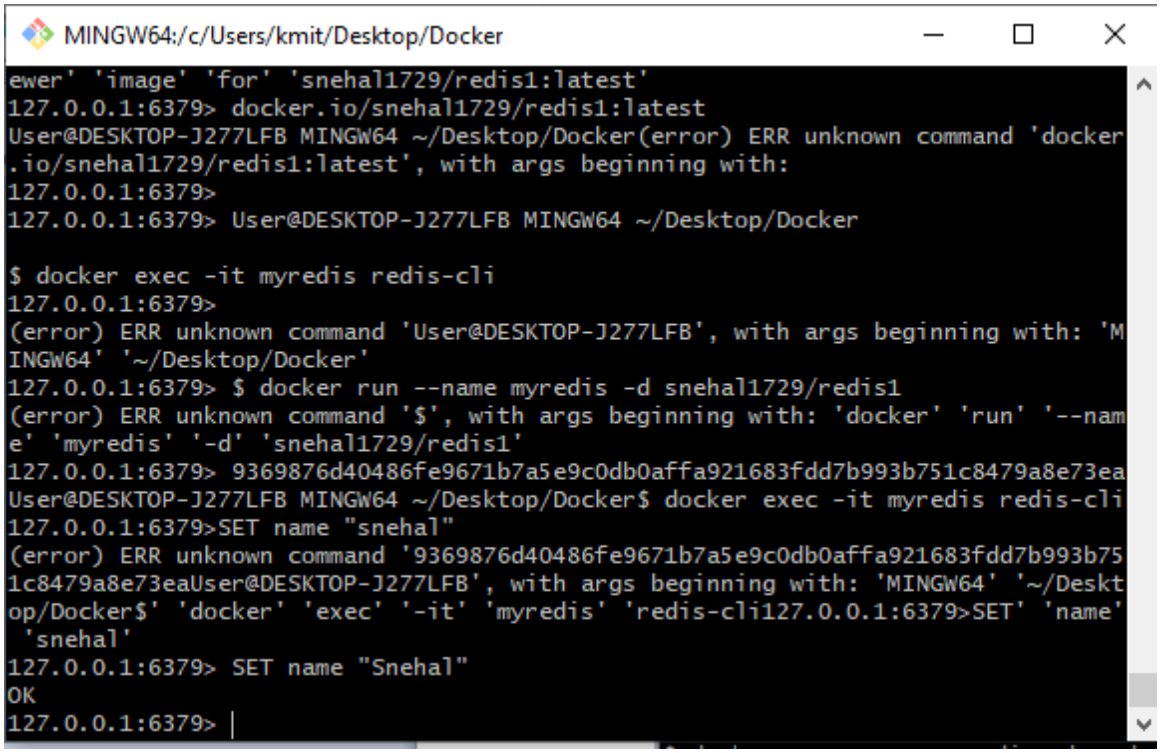
 Opens the Redis command-line interface (like a terminal) inside the running container myredis.




**17. SET name "Abcdef"**

What it does:

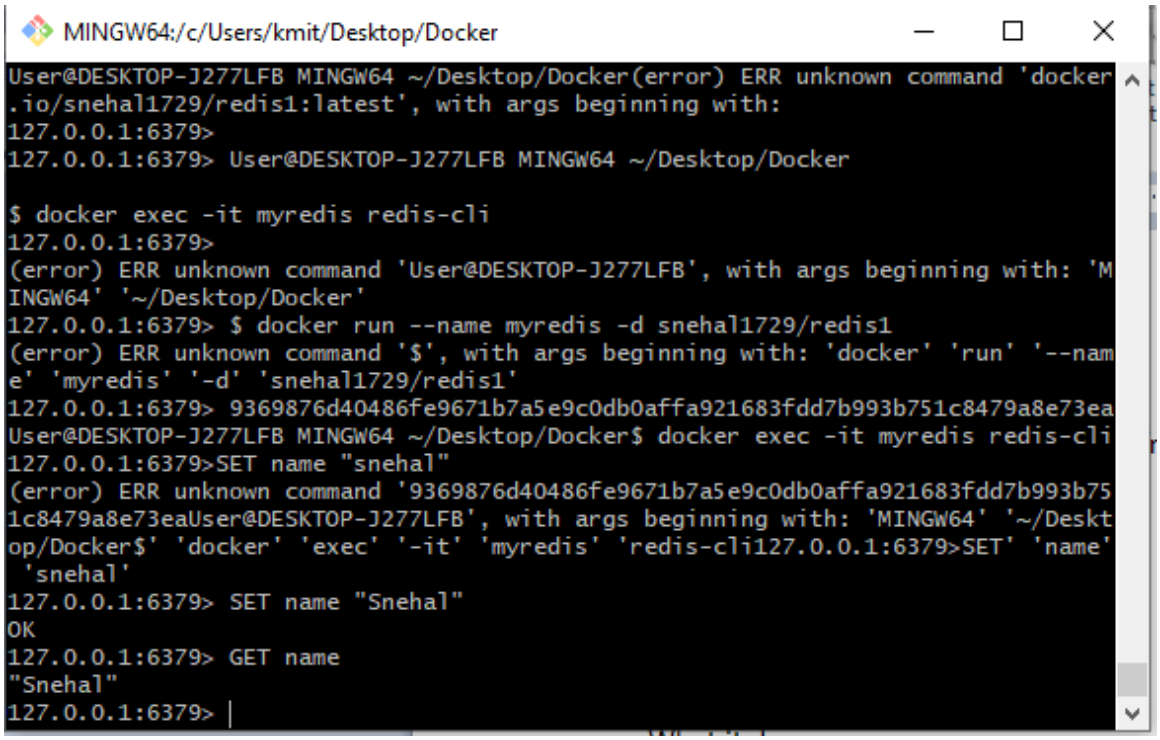
 Saves a key-value pair in Redis (key = name, value = Abcdef).




**18. GET name**

What it does:

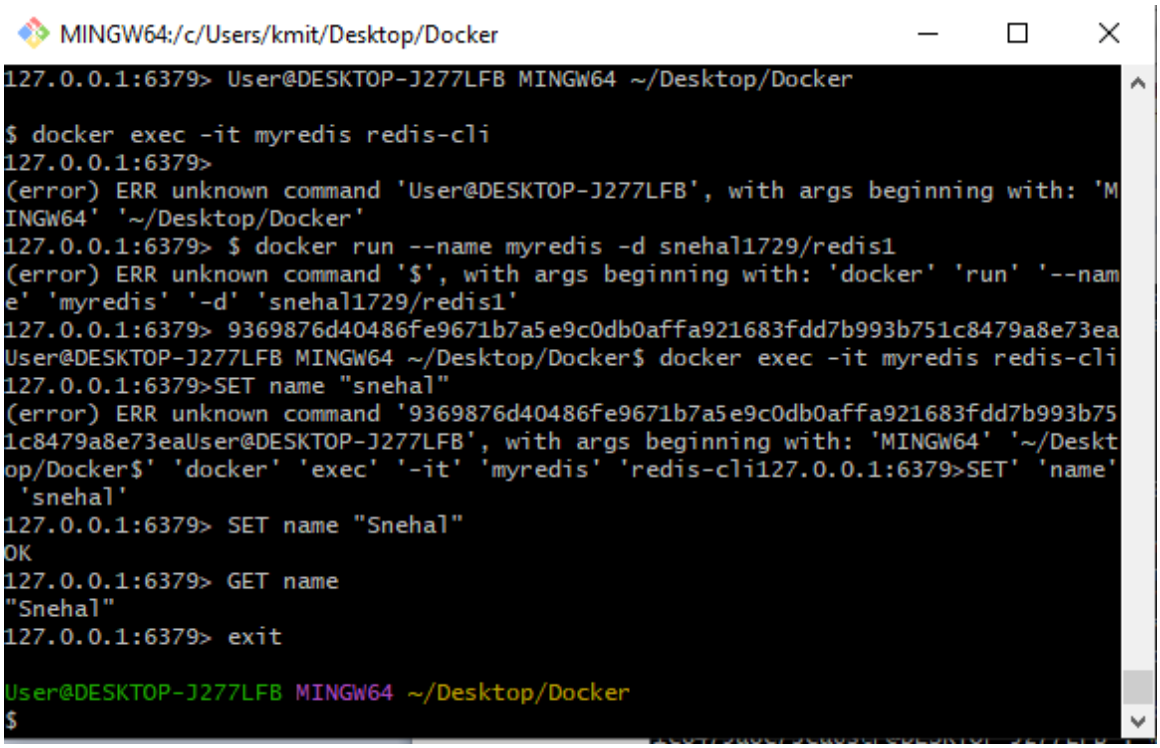
 Retrieves the value of the key name from Redis (it will return "Abcdef").



**19. exit**

What it does:

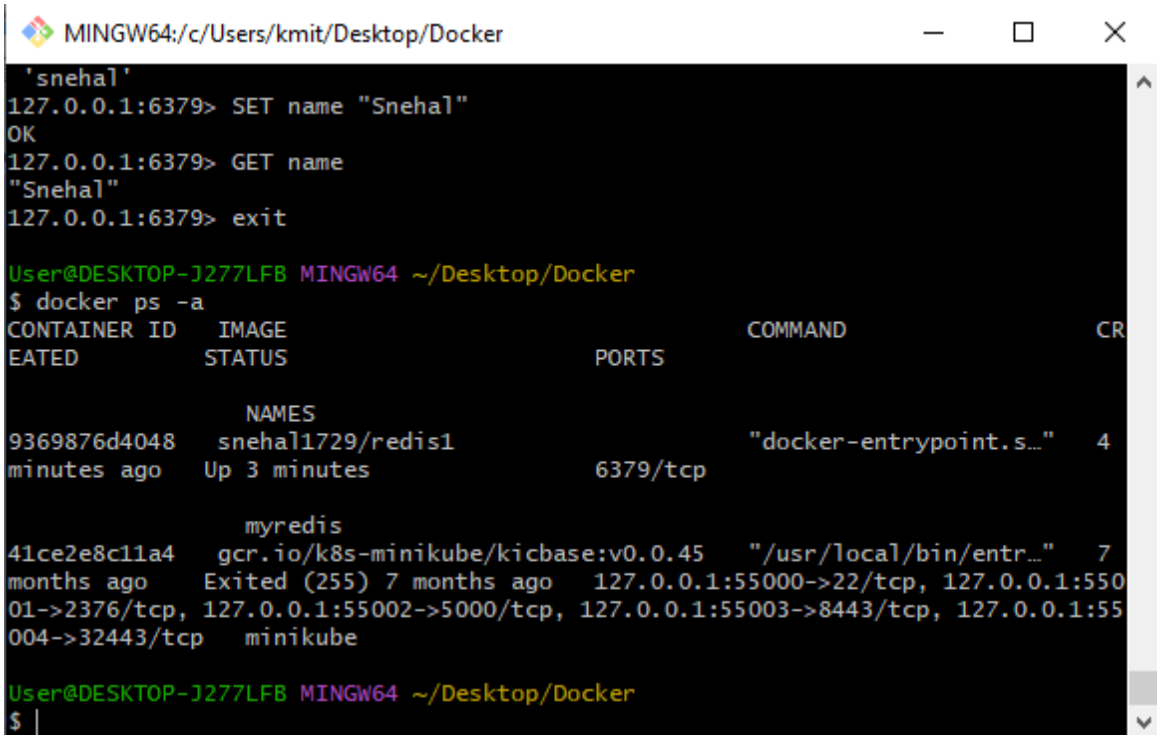
 Exits the Redis CLI.




**20. docker ps -a**

What it does:

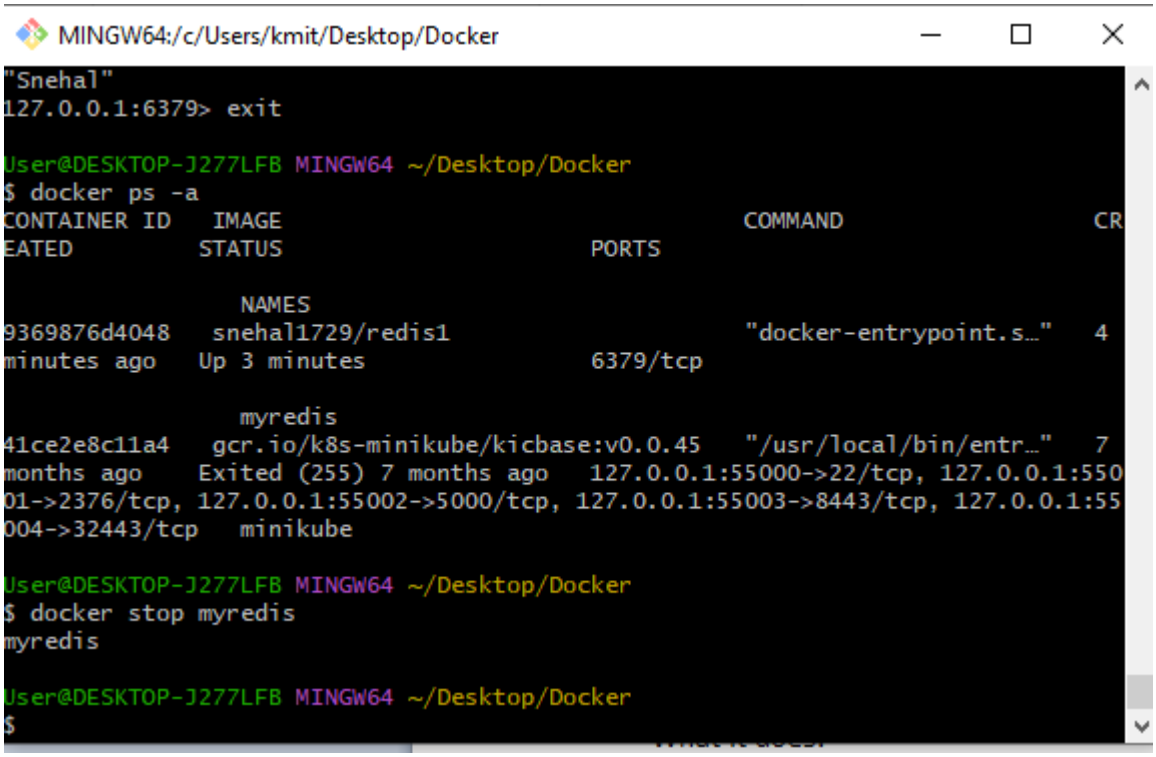
 Shows all containers again to check their status.




**21. docker stop myredis**

What it does:

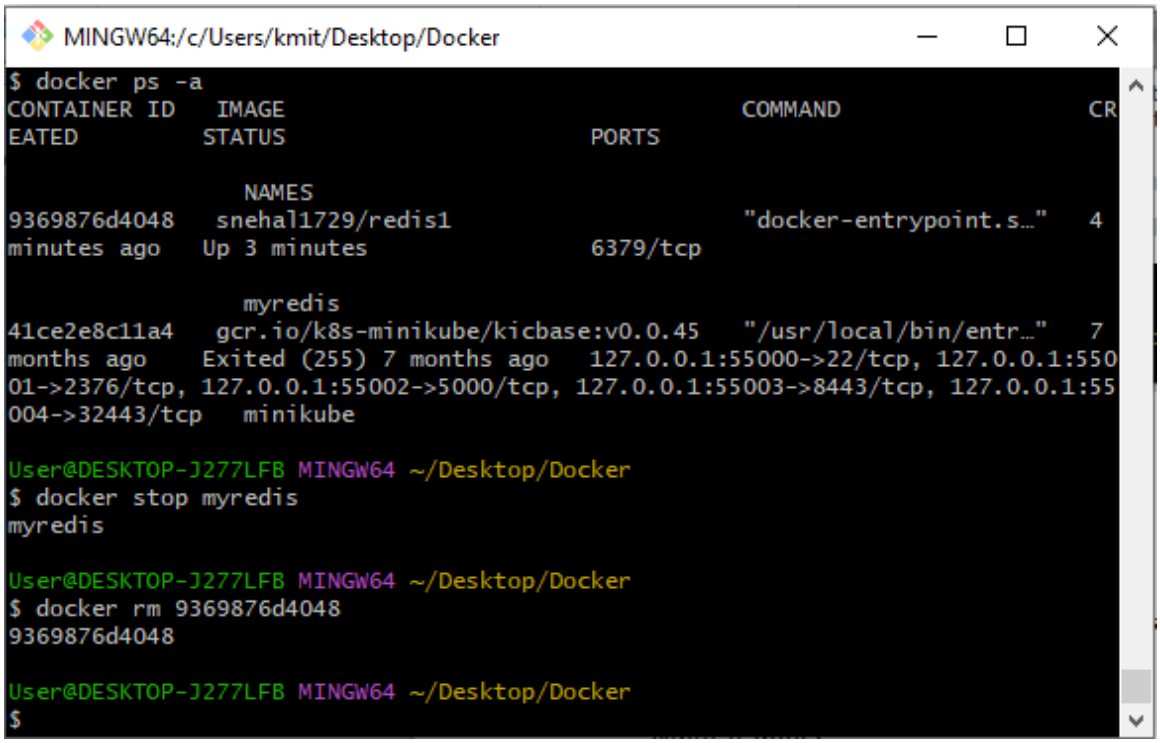
- Stops the container myredis.



**22. docker rm 50a6e4a9c326**

What it does:

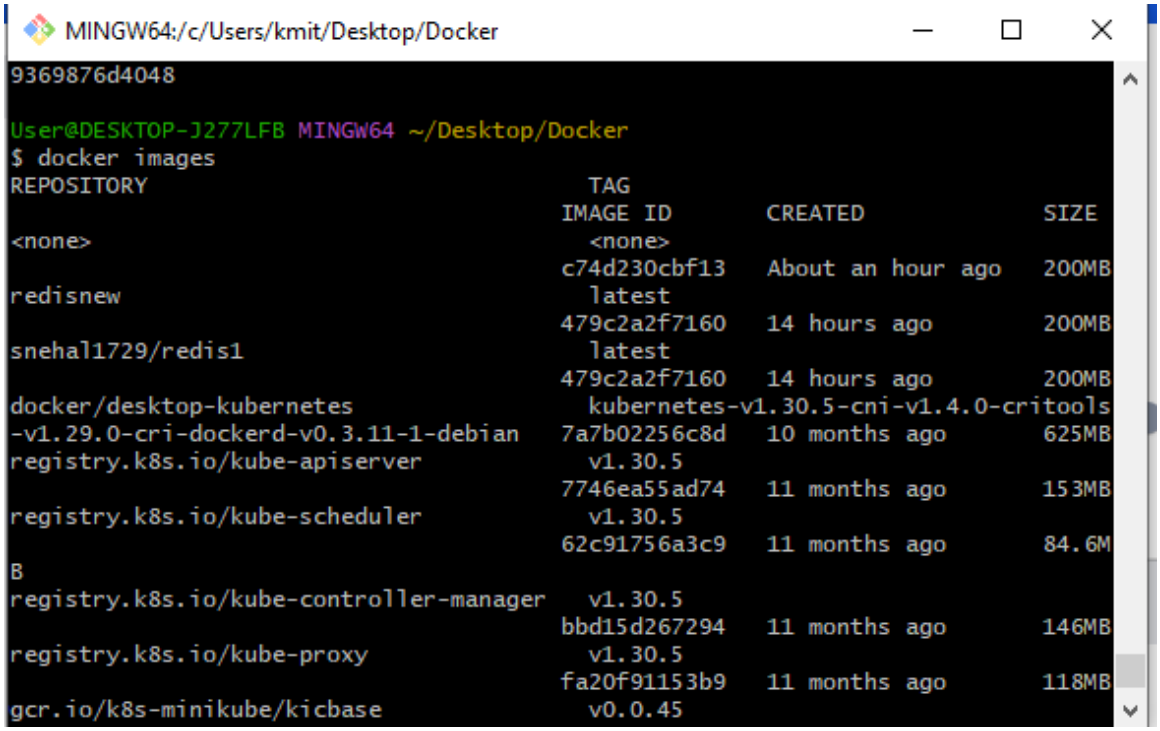
 Deletes the container with ID 50a6e4a9c326.




**23. docker images**

What it does:

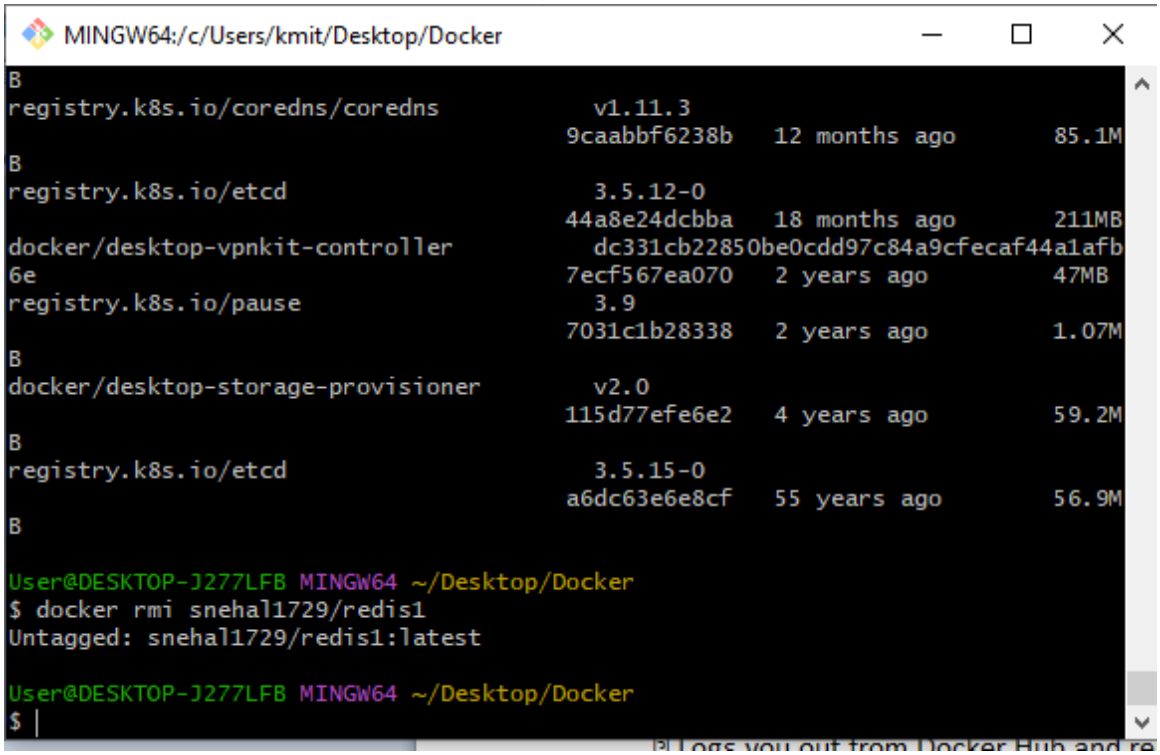
 Lists all images again to confirm which ones remain.




**24. docker rmi snehal1729/redis1**

What it does:

 Deletes the image budarajumadhurika/redis1 again.



**Step 3: Remove Login Credentials (Optional)**

If you no longer need to be logged in, you can log out:

docker logout

What It Does:

 Logs you out from Docker Hub and removes your stored credentials.

## Scenario based Questions:

1.Your application is running inside Docker, but you're not sure if the container is active.
Which command helps you check running containers?

ANS. docker ps

2. You notice one of your containers is consuming high CPU and want to stop it.
How do you stop a running container named web_app?

ANS. docker stop web_app

3. You've written a Dockerfile in your project directory and want to build an image named myapi.

What Docker CLI command do you use?

ANS. docker build -t myapi .

4. You want to run your web_app container and expose its internal port 5000 on host port 8080.
Which command should you use?

ANS. docker run -p 8080:5000 web_app

5. Your container is running, but you want to go inside and debug it using a shell. What CLI command helps you enter the container's shell?

ANS. docker exec -it <container_name_or_id> bash

6. You've built a few test images and want to delete one called old_api. How would you do this?

ANS. docker rmi old_api

7. Your app crashed in the container. You need to check the logs. How do you check this?

ANS. docker logs <container_name_or_id>

8. You want to start a container and let it run in the background. Which command should you use?

ANS. docker run -d <image_name>

9. You're not sure which container is using port 3000. How do you check?

ANS. docker ps --filter "publish=3000"

10. You built an image but forgot to tag it. How do you tag it now?

ANS. docker tag <existing_image_id_or_name> <new_repo>:<new_tag>

11. You want to export an image to a .tar file for offline transfer. How do you export it?

ANS. docker save -o image.tar <image_name_or_id>

12. You want to restart a container automatically if it crashes. What Docker CLI command do you use?

ANS. docker run --restart unless-stopped <image_name>

13. You want to see how much CPU/RAM each container uses. What Docker CLI command do you use? How to Control or Limit RAM in Docker? How to Measure Actual Memory Usage?

ANS. docker stats

docker run -m 512m <image>

14. You want your container to run a shell command before starting the app. What do you include in docker file?

ANS. CMD /bin/sh -c "your-command && exec your-app"

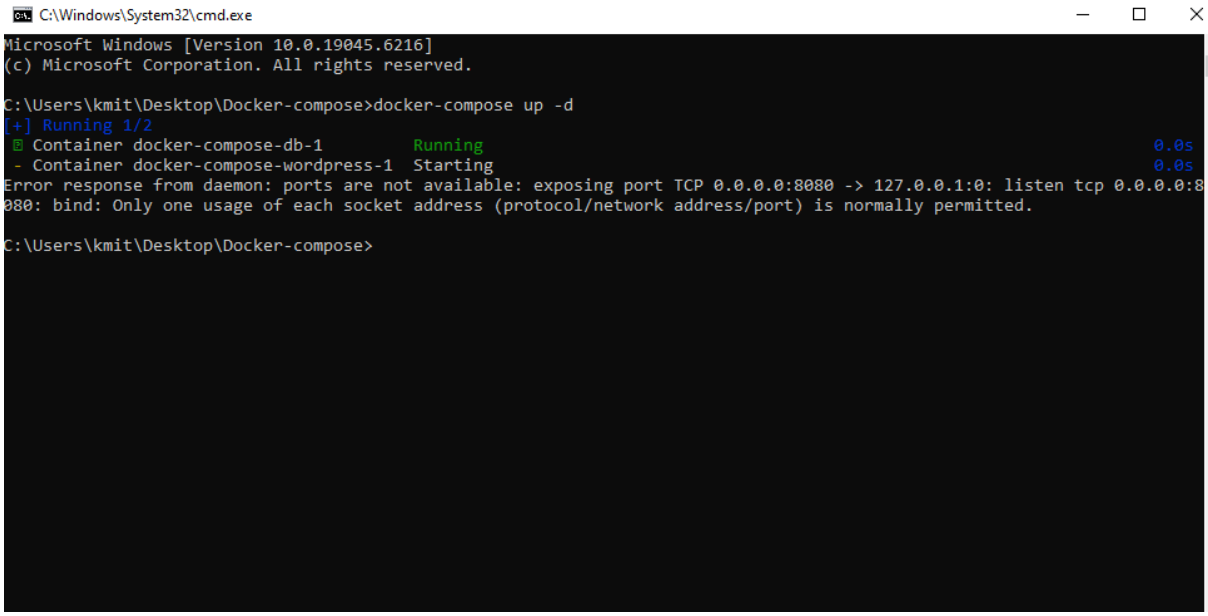
15. You want to create a custom Nginx image with your static files. What do you do?

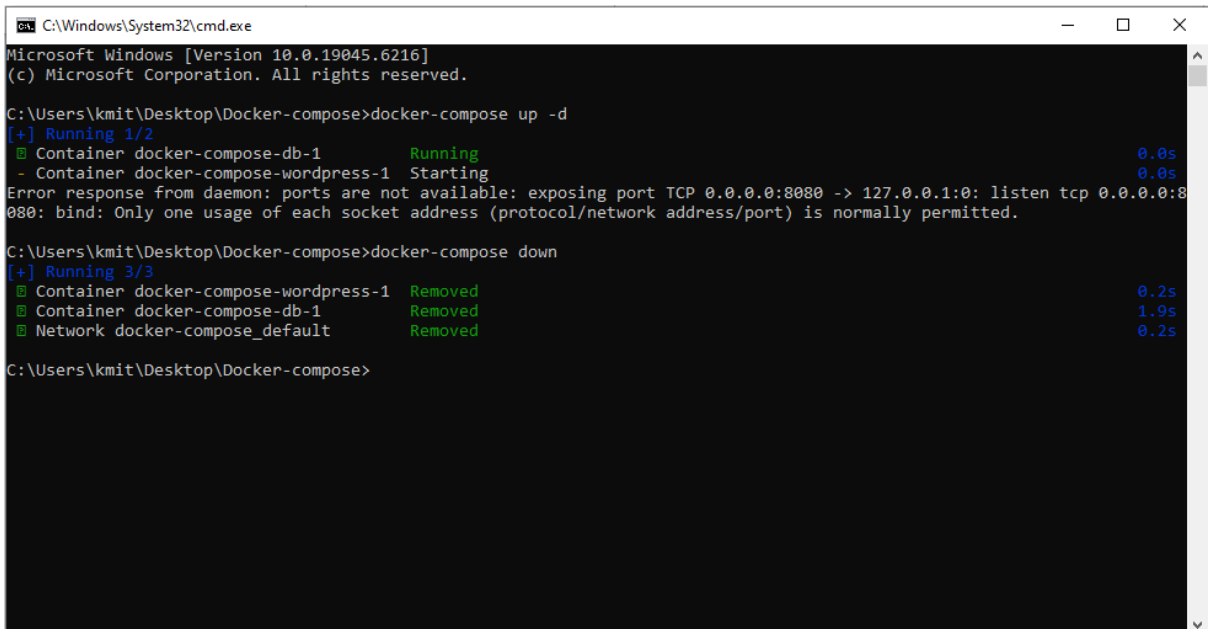
ANS. FROM nginx:latest

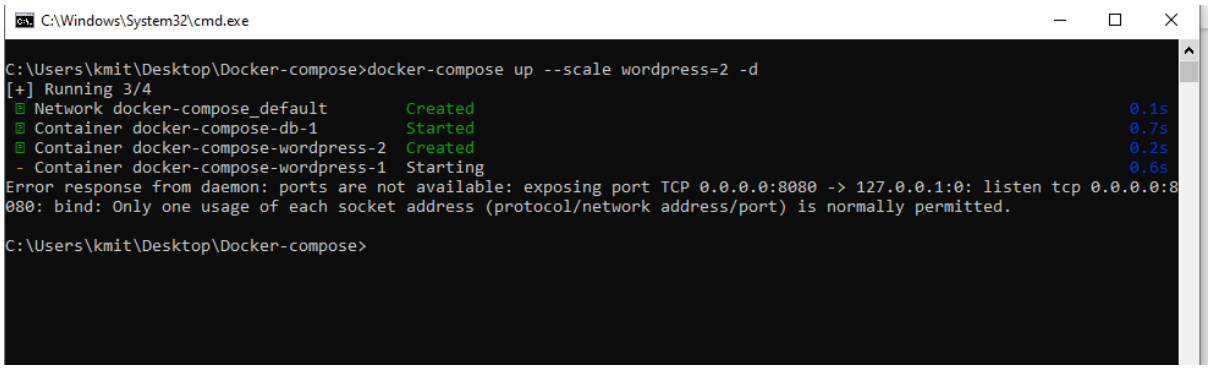
COPY ./static /usr/share/nginx/html

**Conclusion:**

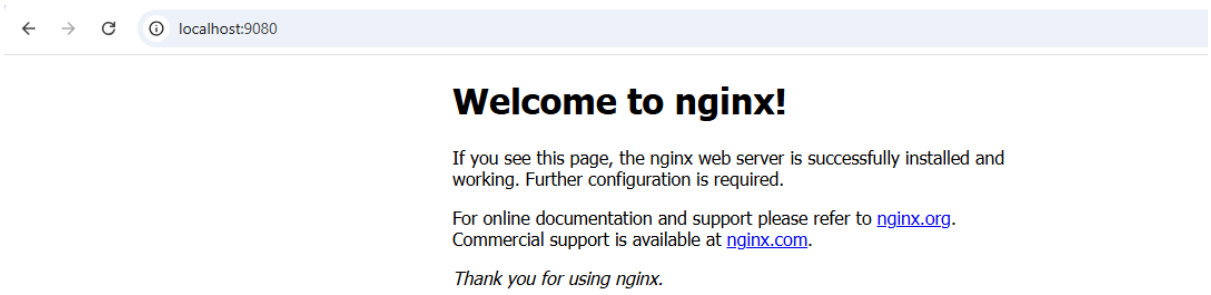
Through this hands-on exploration of Docker commands and Dockerfile usage, students will get a clear picture about the end-to-end container workflow—from pulling images, creating and running containers, to customizing them with Dockerfiles. By practicing key commands such as docker build, docker run, docker ps, and docker exec, students will gain practical skills in managing containerized environments. The Dockerfile exercise reinforced the importance of layered builds, efficient image creation, and environment reproducibility. Overall, these activities strengthened the understanding of containerization principles, enabling building, testing and deploying applications in isolated, consistent environments.


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.
C:\Users\kmit\Desktop\Docker-compose>docker-compose up -d
[+] Running 1/2
Container docker-compose-db-1 Running 0.0s
Container docker-compose-wordpress-1 Starting 0.0s
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 127.0.0.1:0: listen tcp 0.0.0.0:8
080: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
C:\Users\kmit\Desktop\Docker-compose>


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.
C:\Users\kmit\Desktop\Docker-compose>docker-compose up -d
[+] Running 1/2
Container docker-compose-db-1 Running 0.0s
Container docker-compose-wordpress-1 Starting 0.0s
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 127.0.0.1:0: listen tcp 0.0.0.0:8
080: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
C:\Users\kmit\Desktop\Docker-compose>docker-compose down
[+] Running 3/3
Container docker-compose-wordpress-1 Removed 0.2s
Container docker-compose-db-1 Removed 1.9s
Network docker-compose_default Removed 0.2s
C:\Users\kmit\Desktop\Docker-compose>


C:\Windows\System32\cmd.exe
C:\Users\kmit\Desktop\Docker-compose>docker-compose up --scale wordpress=2 -d
[+] Running 3/4
Network docker-compose_default Created 0.1s
Container docker-compose-db-1 Started 0.7s
Container docker-compose-wordpress-2 Created 0.2s
Container docker-compose-wordpress-1 Starting 0.6s
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:8080 -> 127.0.0.1:0: listen tcp 0.0.0.0:8
080: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
C:\Users\kmit\Desktop\Docker-compose>

```
PS C:\Users\kmit\Desktop\Compose-lab> docker compose up -d
time="2025-08-26T10:42:20+05:30" level=warning msg="C:\\Users\\kmit\\Desktop\\Compose-lab\
\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remov
e it to avoid potential confusion"
[+] Running 2/2
 ✓ Container compose-lab-db-1   Running                                    0.0s
 ✓ Container compose-lab-web-1  Started                                    1.1s
PS C:\Users\kmit\Desktop\Compose-lab>
```




**Transfer it to another machine with Docker Compose installed.**

```
PS C:\Users\kmit\Desktop\Compose-lab> docker compose up -d
time="2025-08-26T10:46:27+05:30" level=warning msg="C:\\Users\\kmit\\Desktop\\Compose-lab\\docker-compose.yml:
t to avoid potential confusion"
[+] Running 3/3
 ✓ Container compose-lab-db-1     Running                                  0.0s
 ✓ Container compose-lab-redis-1  Running                                  0.0s
 ✓ Container compose-lab-web-1    Running                                  0.0s
PS C:\Users\kmit\Desktop\Compose-lab>
```

```
C:\Users\kmit\Desktop\Compose-lab>docker compose ps
time="2025-08-26T10:47:39+05:30" level=warning msg="C:\\Users\\kmit\\Desktop\\Compose-lab\\docker-compose.yml: the attribute `version` is obsolete,
please remove it to avoid potential confusion"
NAME                IMAGE          COMMAND                  SERVICE   CREATED         STATUS         PORTS
compose-lab-db-1    postgres:15    "docker-entrypoint.s…"   db        11 minutes ago  Up 11 minutes  5432/tcp
compose-lab-redis-1 redis:alpine   "docker-entrypoint.s…"   redis     2 minutes ago   Up 2 minutes   6379/tcp
compose-lab-web-1   nginx:latest   "/docker-entrypoint.…"   web       5 minutes ago   Up 5 minutes   0.0.0.0:9080->80/tcp, [::]:9080->80/tcp
```

**Check that Nginx and Postgres work there as well.**

```
10
    ▷ Run Service
11  db:
12    image: postgres:15
13    environment:
14      POSTGRES_USER: demo
15      POSTGRES_PASSWORD: demo
16      POSTGRES_DB: demo_db
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS C:\Users\kmit\Downloads\compose-lab> docker compose up -d
time="2025-08-26T10:55:15+05:30" level=warning msg="C:\\Users\\kmit\\Downloads\\compose-la
b\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please rem
ove it to avoid potential confusion"
[+] Running 3/3
 ✓ Container compose-lab-db-1     Running                    0.0s
 ✓ Container compose-lab-redis-1  Running                    0.0s
 ✓ Container compose-lab-web-1    Started                    2.1s
PS C:\Users\kmit\Downloads\compose-lab>
```

localhost:8081

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

*Thank you for using nginx.*

**All services run correctly on the new host.**

```
PS C:\Users\kmit\Downloads\compose-lab> docker compose up -d
[+] Running 5/5
 ✓ Network compose-lab_app-net    Created                    0.1s
 ✓ Volume "compose-lab_db-data"   Created                    0.0s
 ✓ Container compose-lab-db-1     Started                    2.5s
 ✓ Container compose-lab-redis-1  Started                    2.4s
 ✓ Container compose-lab-web-1    Started                    1.9s
PS C:\Users\kmit\Downloads\compose-lab>
```

```yaml
docker-compose.yml
services:
  db:
    environment:
      POSTGRES_USER: demo
      POSTGRES_PASSWORD: demo
      POSTGRES_DB: demo_db
    volumes:
      - db-data:/var/lib/postgresql/data
    networks:
      - app-net
    ports:
      -"9090:4060"

  ▷Run Service
  redis:
    image: redis:alpine
    networks:
      - app-net
```

## . Insert some data into Postgres

## Then

```
                                                                    2.4s
 ✓ Container compose-lab-web-1    Started
                                                                    1.9s
PS C:\Users\kmit\Downloads\compose-lab> docker compose down
[+] Running 4/4
 ✓ Container compose-lab-redis-1  Removed                           0.4s
 ✓ Container compose-lab-web-1    Removed                           0.4s
 ✓ Container compose-lab-db-1     Removed                           0.4s
 ✓ Network compose-lab_app-net    Removed                           0.2s
PS C:\Users\kmit\Downloads\compose-lab>
```

```
PS C:\Users\kmit\Downloads\compose-lab> docker compose up -d
[+] Running 3/3
 ✓ Container compose-lab-redis-1  Running                           0.0s
 ✓ Container compose-lab-db-1     Running                           0.0s
 ✓ Container compose-lab-web-1    Running                           0.0s
PS C:\Users\kmit\Downloads\compose-lab> docker compose ps
NAME                  IMAGE          COMMAND                  SERVICE   CREATED              STATU
S              PORTS
```

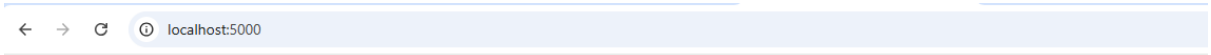
```
PS C:\Users\kmit\Downloads\compose-lab> docker compose up -d
time="2025-08-26T11:05:38+05:30" level=warning msg="Found orphan containers ([compose-lab-redis-1]) for th
is project. If you removed or renamed this service in your compose file, you can run this command with the
 --remove-orphans flag to clean it up."
[+] Running 2/2
 ✓ Container compose-lab-db-1   Running                             0.0s
 ✓ Container compose-lab-web-1  Started                             1.1s
PS C:\Users\kmit\Downloads\compose-lab>
```

```
PS C:\Users\kmit\Downloads\compose-lab> docker compose ps
NAME                   IMAGE          COMMAND                  SERVICE   CREATED              STATU
S              PORTS
compose-lab-db-1       postgres:15    "docker-entrypoint.s…"   db        50 seconds ago       Up 49
 seconds       0.0.0.0:9090->5432/tcp, [::]:9090->5432/tcp
compose-lab-redis-1    redis:alpine   "docker-entrypoint.s…"   redis     7 minutes ago        Up 7
minutes        6379/tcp
compose-lab-web-1      nginx:latest   "/docker-entrypoint.…"   web       About a minute ago   Up Ab
out a minute   0.0.0.0:8081->80/tcp, [::]:8081->80/tcp
PS C:\Users\kmit\Downloads\compose-lab>
```

```
#10 naming to docker.io/library/compose-lab-web:latest done
#10 unpacking to docker.io/library/compose-lab-web:latest
#10 unpacking to docker.io/library/compose-lab-web:latest 0.4s done
#10 DONE 1.8s

#11 resolving provenance for metadata file
#11 DONE 0.0s
[+] Running 4/4
 ✓ compose-lab-web                Built                                                    0.0s
 ✓ Container compose-lab-db-1     Running                                                  0.0s
 ✓ Container compose-lab-redis-1  Running                                                  0.0s
 ✓ Container compose-lab-web-1    Started                                                  2.2s
PS C:\Users\kmit\Downloads\compose-lab>
```

```
PS C:\Users\kmit\Downloads\compose-lab> docker compose ps
>>
NAME                IMAGE             COMMAND                  SERVICE   CREATED          STATUS          PORTS
compose-lab-db-1    postgres:15       "docker-entrypoint.s…"   db        6 minutes ago    Up 6 minutes    0.0.0.0:9090->5432/tcp, [::]:9090->5432
/tcp
compose-lab-redis-1   redis:alpine      "docker-entrypoint.s…"   redis     12 minutes ago   Up 12 minutes   6379/tcp
compose-lab-web-1   compose-lab-web   "python app.py"          web       56 seconds ago   Up 54 seconds   0.0.0.0:5000->5000/tcp, [::]:5000->5000
/tcp
PS C:\Users\kmit\Downloads\compose-lab>
```

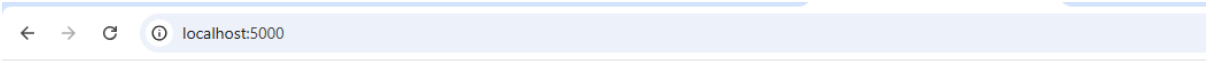



Hello from Flask + Docker!

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                                    docker

 ✓ compose-lab-web                Built                                                          0.0s
 ✓ Container compose-lab-redis-1  Running                                                        0.0s
 ✓ Container compose-lab-db-1     Running                                                        0.0s
 ✓ Container compose-lab-web-1    Recreated                                                     12.5s
Attaching to db-1, redis-1, web-1
web-1  |  * Serving Flask app 'app'
web-1  |  * Serving Flask app 'app'
web-1  |  * Debug mode: off
web-1  | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
web-1  |  * Running on all addresses (0.0.0.0)
web-1  |  * Running on http://127.0.0.1:5000
web-1  |  * Running on http://172.23.0.4:5000
web-1  | Press CTRL+C to quit
web-1  |  * Debug mode: off
web-1  | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
web-1  |  * Running on all addresses (0.0.0.0)
web-1  |  * Running on http://127.0.0.1:5000
web-1  |  * Running on http://172.23.0.4:5000
web-1  | Press CTRL+C to quit

 View in Docker Desktop   View Config   Enable Watch
```




Hello Docker Compose!

# LAB ACTION PLAN FOR WEEK 7

## Objective:

To enable students to:

- Understand the structure and lifecycle of a Maven project.
- Build and package Java and Web applications using Maven.
- Add dependencies using **pom.xml**, compile and test using plugins.
- Resolve errors and conflicts arising from dependency mismatches.
- Work with parent and multi-module Maven projects.
- Generate executable JARs and deployable WARs using Maven.

Students must **document observations**, include **screenshots of executions**, and answer **scenario-based questions** after completing the tasks.

## 1. Understanding Maven Project Structure

Maven standardizes the project structure for both Java and web-based applications. src/

```
├── main/
│   ├── java/        → Application source code
│   └── resources/   → Configuration files like config.properties
└── test/
    ├── java/        → Unit test source code
    └── resources/   → Test resources
```

The compiled files and reports are generated in the **`target/`** directory after a successful build.

## 2. Steps to Perform Maven Build and Testing

**----mvn clean install**

- **`clean`:** Deletes the previous build (**`target/`** folder)
- **`install`:** Builds, tests, and installs the package to local **`.m2`** repository

After execution:

- **`target/`** folder contains compiled **`.class`** files and the final **`.jar`** or **`.war`**

- Artifact is stored in:
  **`~/.m2/repository/groupId/artifactId/version/`**

Add a Dependency (e.g., Gson):

In **`pom.xml`**:

*<dependency>*

*<groupId>com.google.code.gson</groupId>*

*<artifactId>gson</artifactId>*

*<version>2.10</version>*

*</dependency>*

After running:

*mvn clean install*

Check:

- Gson JAR is downloaded to **`.m2/repository/com/google/code/gson/gson/`**
- If version is wrong or not found → **`BUILD FAILURE`** with dependency resolution error.

## Creation of Maven Java Project

**Step 1.** Open Eclipse IDE

└— 1.1. 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

**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.

## 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

**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.

**Note:-Now push yours Maven java project and Maven Web Project into your github**

## 3. Configure Java Version via Compiler Plugin

In **`pom.xml`**:

*<plugin>*

*<groupId>org.apache.maven.plugins</groupId>*

*<artifactId>maven-compiler-plugin</artifactId>*

*<version>3.11.0</version>*

*<configuration>*

*<source>17</source>*

*<target>17</target>*

*</configuration> </plugin>*

Check:

- Run `mvn package`
- Inspect generated JAR: *jar tf target/myapp.jar*

## 4. JUnit Testing and Reports

Place test files in `src/test/java`. Example:

```
public class AppTest {
    @Test
    public void testSum() {
        assertEquals(5, 2 + 3);
    }
}
```

Run:

mvn test

Check:

- `target/test-classes/` → compiled test `.class` files
- `target/surefire-reports/` → `.txt` or `.xml` test results
  If test fails → corresponding log and failure trace shown

## 5. Handling Errors in pom.xml

- Typos in version numbers or missing repositories cause `BUILD FAILURE`
- Maven shows precise error in console
- Fix the dependency tag → re-run `mvn clean install`

## 6. Adding Resource Files

Put config.properties inside:

src/main/resources/config.properties

To read:

```
InputStream input =
getClass().getClassLoader().getResourceAsStream("config.properties");
```

After build, check target/classes/ → file should exist there.

## 7. Multi-Module and Parent Projects

Structure:

```
parent/
├── pom.xml (packaging: pom)
├── core/
│   └── pom.xml
└── web/
    └── pom.xml
```

- Parent `pom.xml` defines `<modules>` and common dependencies
- Each submodule builds independently into its `target/` directory

## 8. Executable JARs

Add to `pom.xml`:

```
<build>
 <plugins>
  <plugin>
   <groupId>org.apache.maven.plugins</groupId>
   <artifactId>maven-jar-plugin</artifactId>
   <configuration>
    <archive>
     <manifest>
      <mainClass>com.example.Main</mainClass>
     </manifest>
    </archive>
   </configuration>
  </plugin>
 </plugins>
</build>
```

Run:

```
mvn package
java -jar target/myapp.jar
```

## 9. Building a WAR File

Create a Maven web project with structure:

```
src/main/webapp/
 └── WEB-INF/web.xml
```

Add:

*<packaging>war</packaging>*

*Command:*

*mvn package*

Generates target/mywebapp.war → deploy on Tomcat server.

**10. Scenario-Based Questions:**

1. **If my error is about:**
   ***Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.8.1:compile***
   ***[ERROR] -> [Help 1]***
   ***[ERROR] To see the full stack trace of the errors, re-run Maven with the -X switch.***
   **How can I resolve it using maven terminal?**
   **ANS: mvn clean compile -X**

2. **A dependency you added is not recognized by the compiler. What steps would you take to confirm it is available in .m2 and listed in dependency tree?**
   **ANS: Check if dependency JAR exists in local repo:**
   **~/.m2/repository/<groupId-path>/<artifactId>/<version>/**

3. **A teammate sends a .patch file for a bug fix. How would you apply it and include it in your Maven build?**
   **Apply the patch in your project root using git**
   **git apply path/to/patchfile.patch**
   **You have multiple JUnit test failures and want to rerun only failed tests. How would you approach this?**
   **ANS: Use Maven's -Dtest**
4. **mvn test -Dsurefire.runOrder=failed**

5. **Your Maven build fails due to "Unsupported class version error." What plugin and configuration would you review?**
   **ANS: Check the maven-compiler-plugin Java version settings in pom.xml**

6. **You need to change your Java application from a WAR to a standalone JAR. What pom.xml changes are needed?**
   **ANS:Modify pom.xml packaging:**
   **<packaging>jar</packaging>**

7. **You are required to change the default build output directory from target/ to build_output/. How would you configure it?**
   **ANS: In pom.xml:**
   **<build>**
   **<directory>build_output</directory>**
   **</build>**

8. **You want to skip tests during the Maven build. What command would you use?**
   **ANS: mvn clean install -DskipTests**

9. **How would you generate a site report (with test coverage, dependency analysis) for a Maven project?**
   **ANS: mvn site**

10. **How do you build a Java project using Maven, and what files are generated in the target/ folder after running mvn clean install?**
    **ANS: mvn clean install**

11. **How does Maven resolve dependency conflicts when two libraries use different versions of the same dependency, and how can you view and manage the dependency tree?**
    **Maven uses nearest-wins strategy.**
    **mvn dependency:tree**
    **Use <dependencyManagement> in parent pom.xml to control versions explicitly.**

12. **How do you write and run a JUnit test in a Maven project, and where are the compiled test classes and reports stored after running mvn test?**
    **ANS: Write test in src/test/java with proper JUnit annotations.**
    **Run tests with:**
    **mvn test**
    **Compiled tests: target/test-classes/**

13. **How can you create an executable JAR with a main method using Maven, and which plugin helps configure this behavior?**
    **ANS: Use the maven-jar-plugin or maven-shade-plugin.**

14. **How do you install and use a custom third-party JAR file in your Maven project, and how can you confirm it's included in the build and classpath?**
    **ANS: mvn install:install-file -Dfile=path/to/jar -DgroupId=com.custom -DartifactId=custom-lib -Dversion=1.0 -Dpackaging=jar**

15. **How do you create a Maven web project that packages into a WAR file, and what is the standard folder structure for such a project?**
    **ANS: In pom.xml**

**<packaging>war</packaging>**

16. **What command do you use to build a WAR file in Maven, where is it generated, and how can you deploy it to a server like Apache Tomcat?**
    **Build WAR:**
    **mvn clean package**
    **WAR generated at:**
    **target/yourproject.war**
    **Deploy by copying WAR to Tomcat's webapps/ folder or via Tomcat Manager.**

17. **How do you add JSTL and servlet-api dependencies in a Maven web project, and why should the servlet API use provided scope instead of compile?**
    **Add JSTL and servlet-api dependencies; why servlet-api is**
    **<dependency>**
    **<groupId>javax.servlet</groupId>**
    **<artifactId>javax.servlet-api</artifactId>**
    **<version>4.0.1</version>**
    **<scope>provided</scope>**
    **</dependency>**

    **<dependency>**
    **<groupId>javax.servlet.jsp.jstl</groupId>**
    **<artifactId>javax.servlet.jsp.jstl-api</artifactId>**
    **<version>1.2.1</version>**
    **</dependency>**

18. **How do you set up a multi-module Maven web project with separate modules for core logic and web interface, and how are these modules built and connected?**
    **ANS: Parent pom.xml with <modules> listing core and web modules.**
    **Core module: logic, packaged as jar.**
    **Web module: depends on core, packaged as war.**
    **mvn clean install**

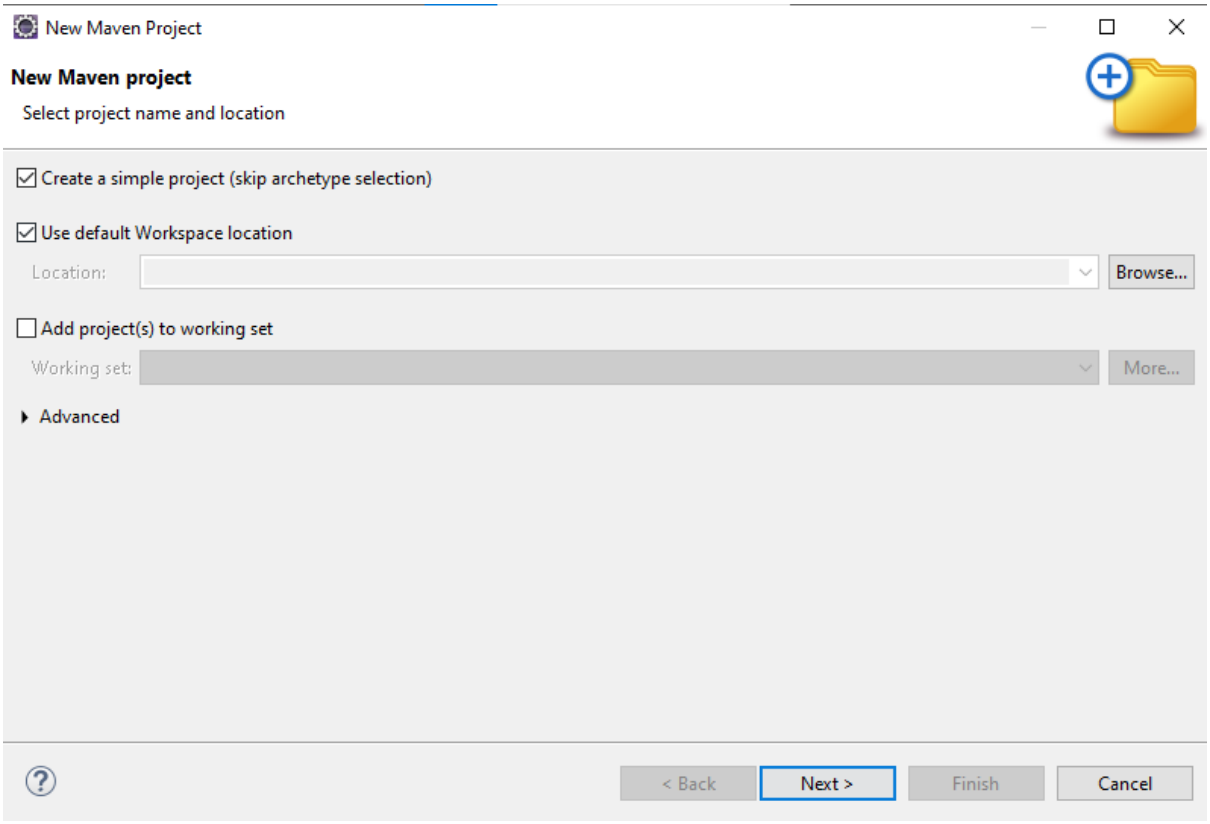
19. **How do you configure a Maven web project, and how does its packaging and execution differ from a traditional WAR-based application?**
    **ans: Uses packaging WAR.**
    **src/main/webapp folder contains web resources.**
    **Built WAR deployed to servlet container.**

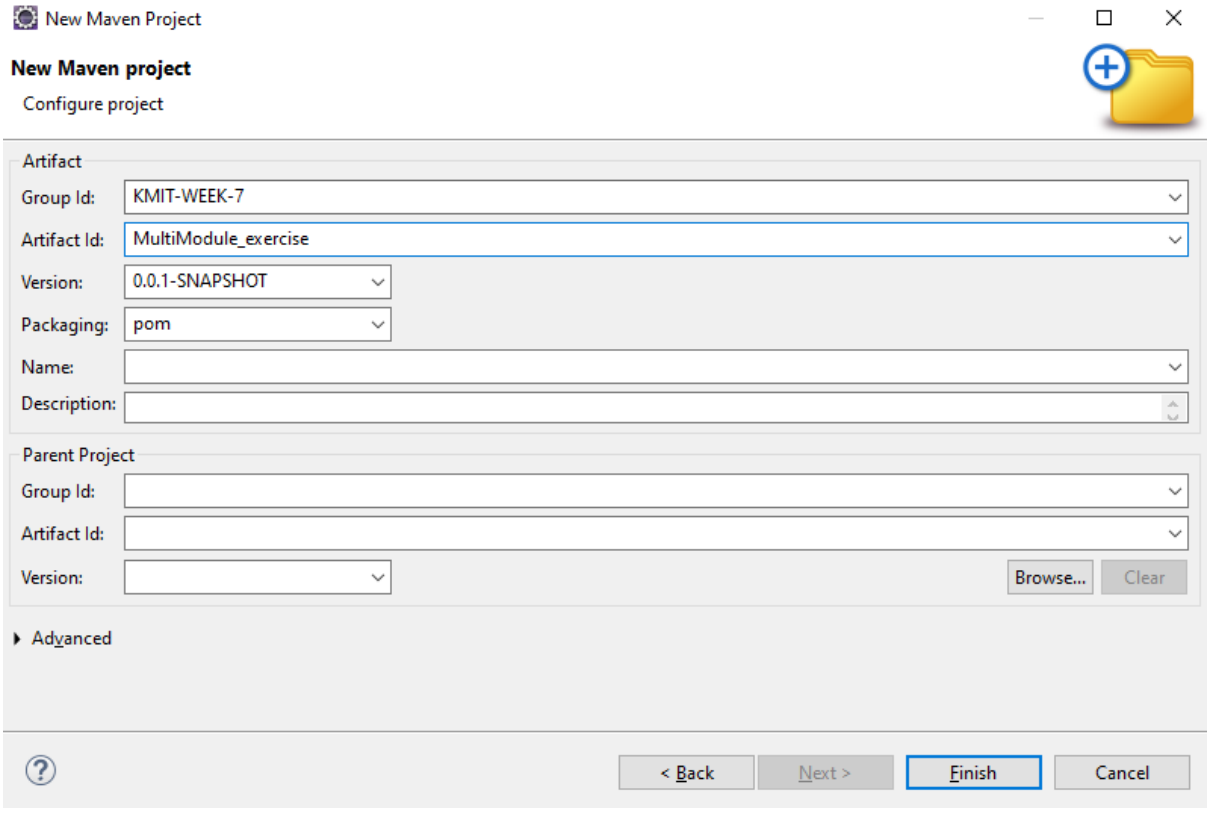
**Can use plugins like tomcat7-maven-plugin for execution.**

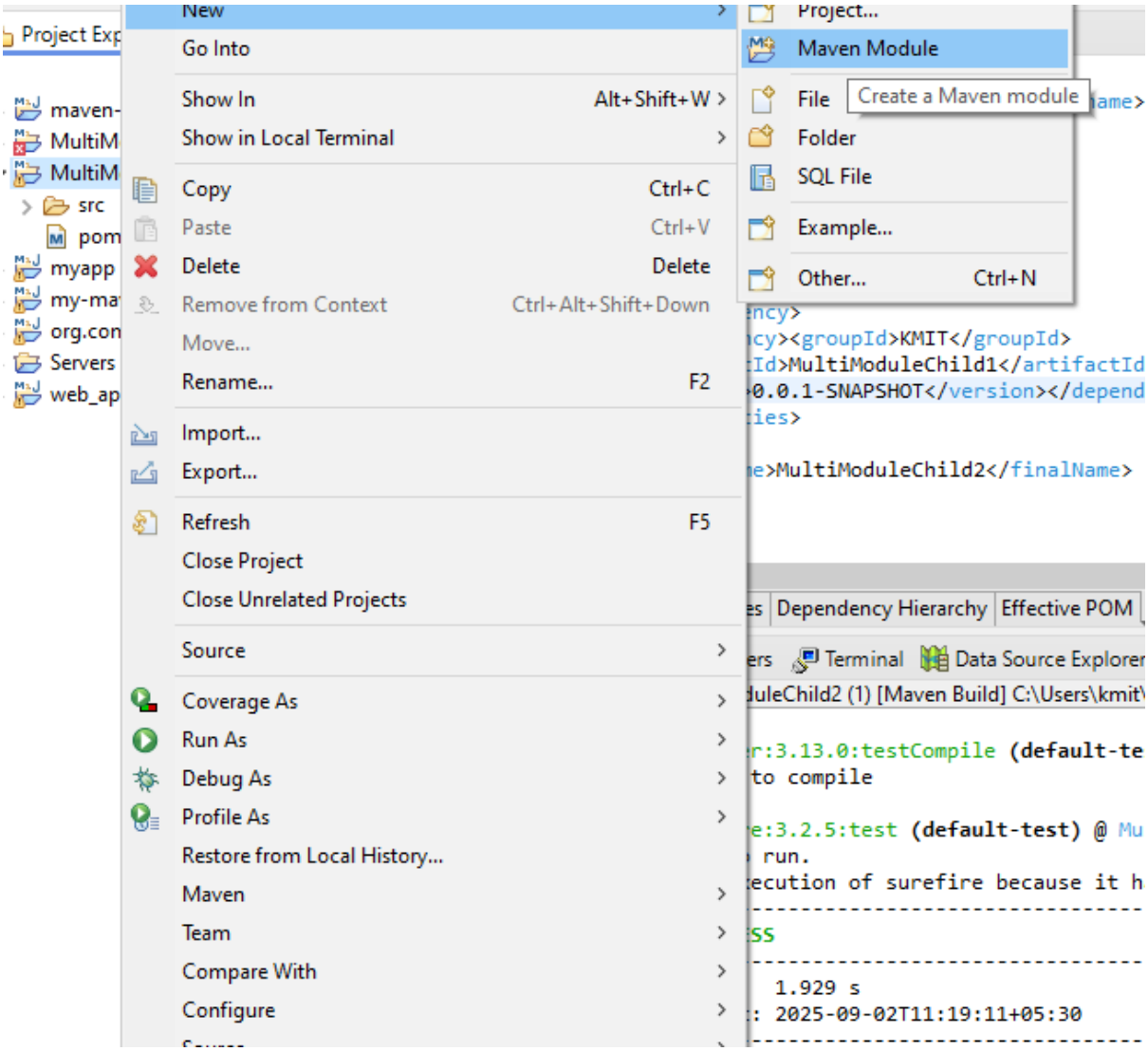
## Conclusion

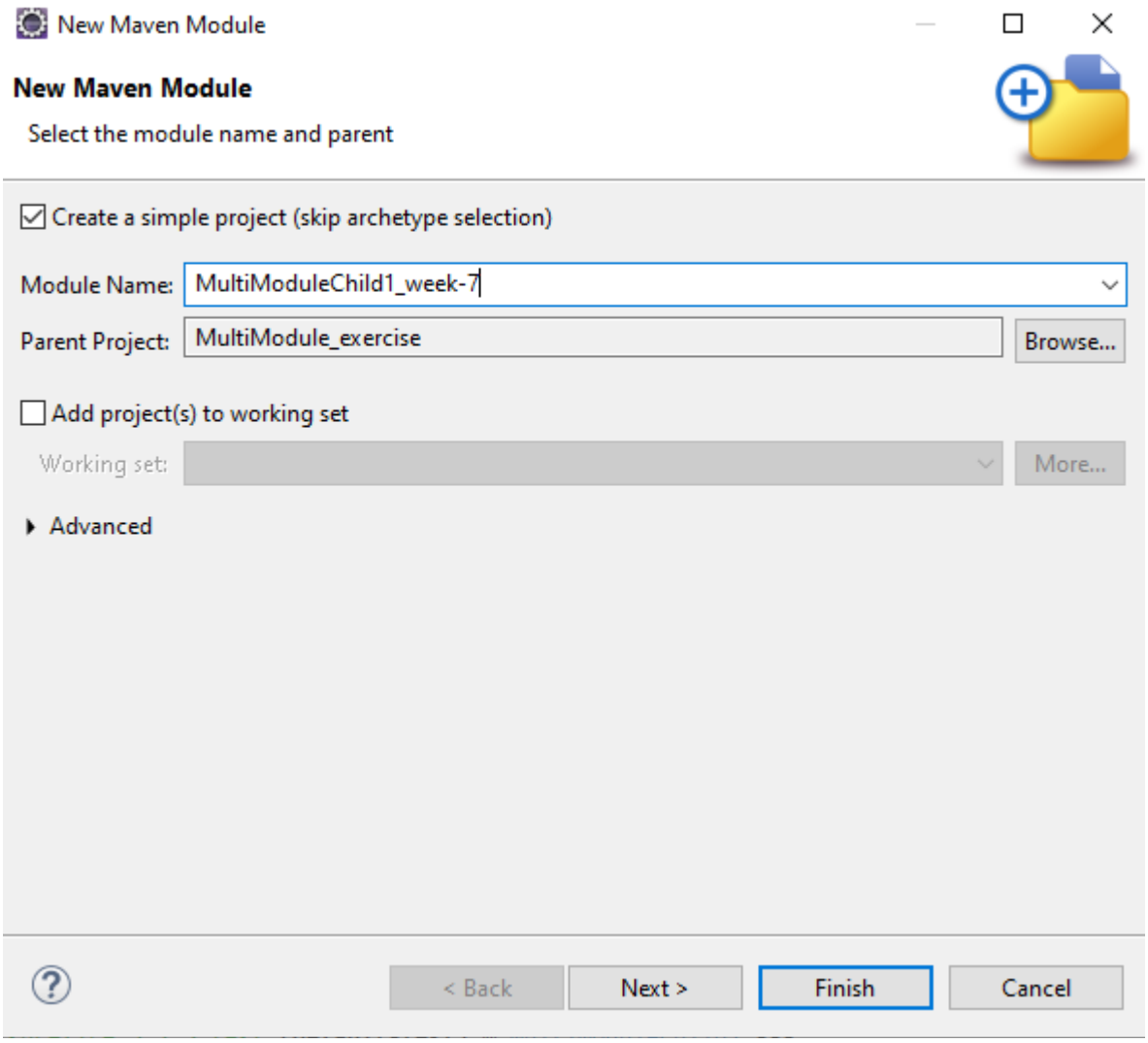
Mastering Maven empowers students to structure projects efficiently, manage complex dependencies, and automate testing and packaging. By practicing real-world scenarios—such as resolving build errors, handling resource files, applying patches, and deploying to servers—students gain valuable hands-on experience aligned with professional software development workflows. Maven not only streamlines builds but also enforces standardization and reusability across projects. Through Maven, students learn efficient project management, dependency control, multi-module integration, and reproducible builds. Understanding Maven's lifecycle, plugin system, and error handling prepares students for professional DevOps and CI/CD workflows. This lab equips students with hands-on experience in packaging, testing, resolving conflicts, and deploying real-world Java and Web applications.

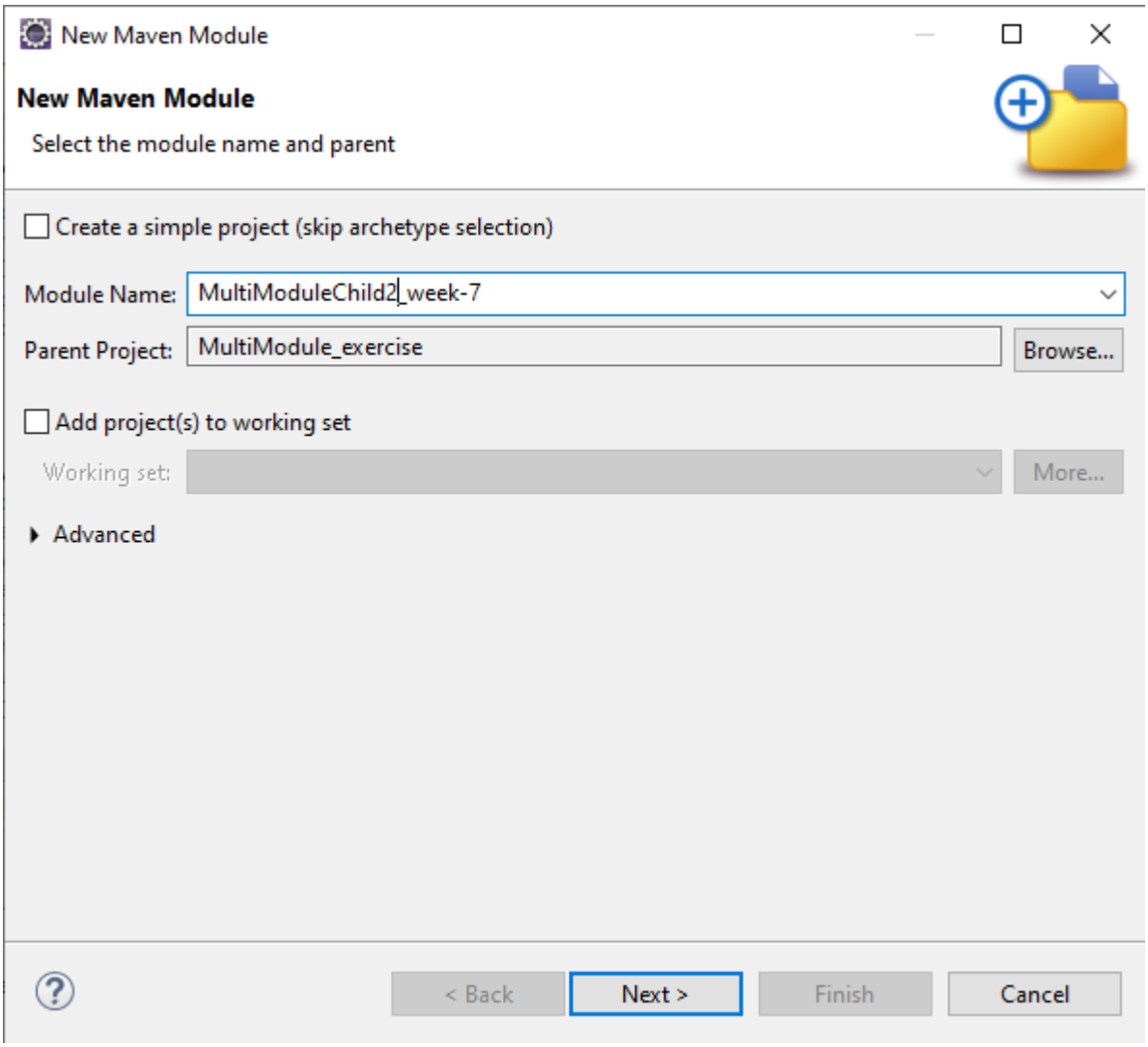
**Multi-Model exercise:**

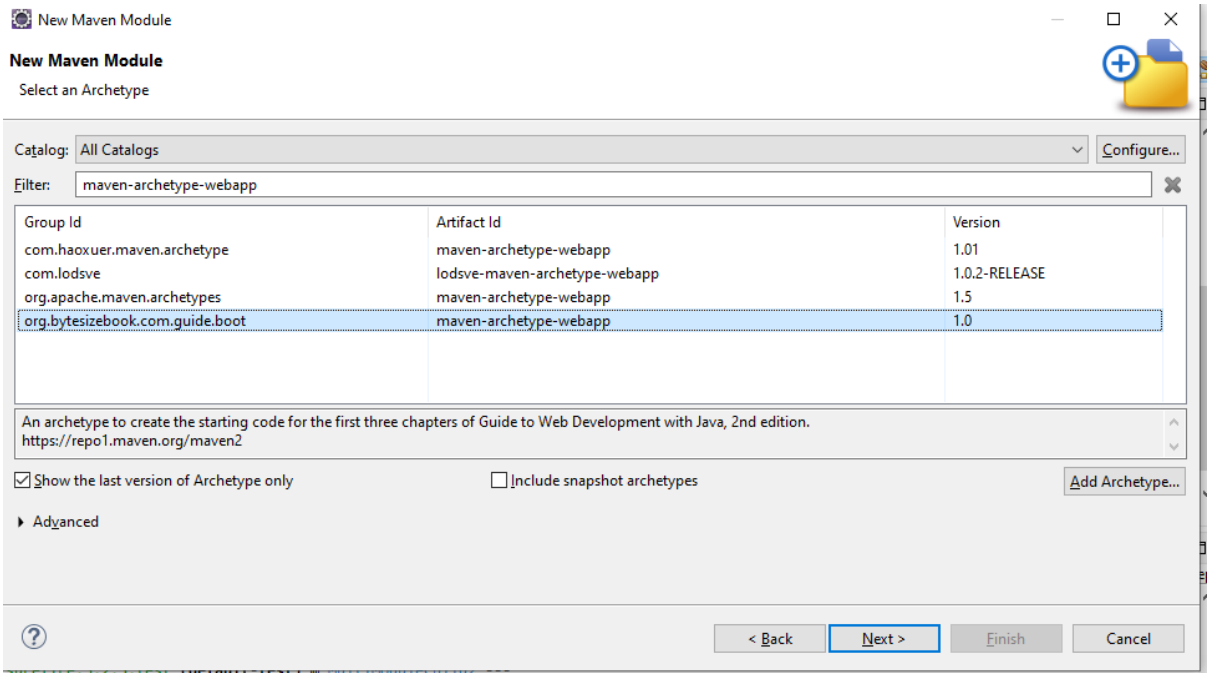


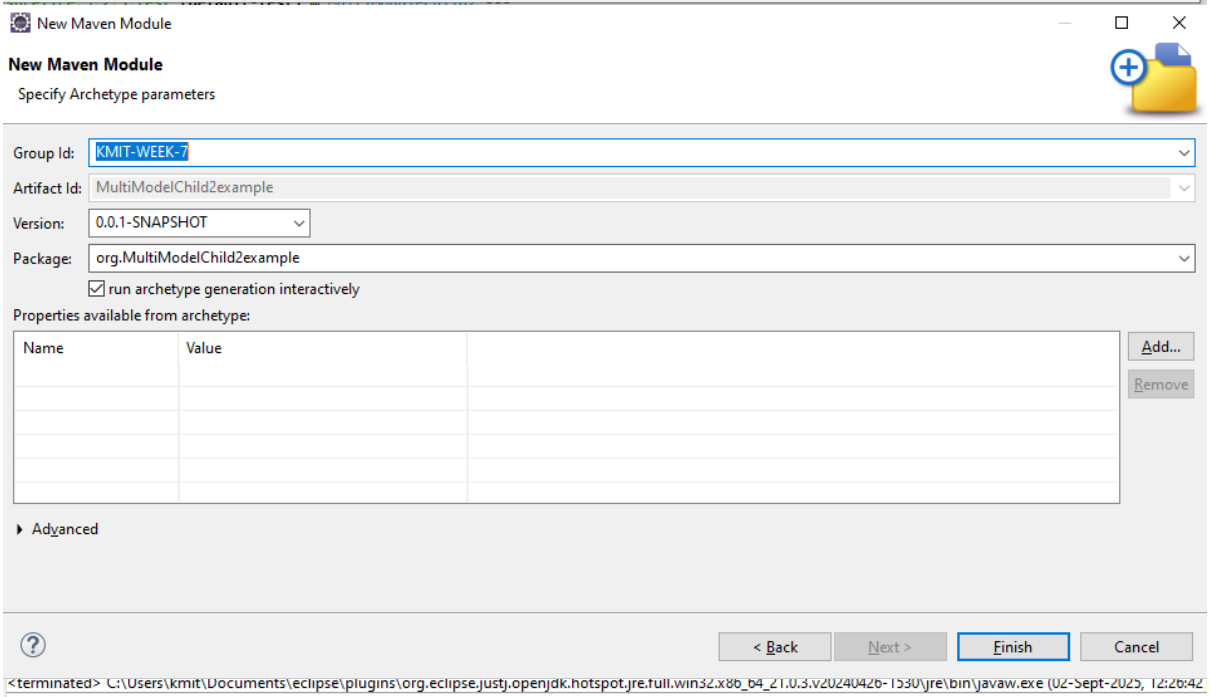




Project Exp
maven-
MultiM
MultiM
src
pom
myapp
my-ma
org.con
Servers
web_ap
New
Go Into
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Move...
Rename... F2
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Source
Coverage As
Run As
Debug As
Profile As
Restore from Local History...
Maven
Team
Compare With
Configure
Project...
Maven Module
File
Create a Maven module
Folder
SQL File
Example...
Other... Ctrl+N
ncy>
cy><groupId>KMIT</groupId>
Id>MultiModuleChild1</artifactId
0.0.1-SNAPSHOT</version></depend
ies>
e>MultiModuleChild2</finalName>
es Dependency Hierarchy Effective POM
ers Terminal Data Source Explorer
duleChild2 (1) [Maven Build] C:\Users\kmit
r:3.13.0:testCompile (default-te
to compile
e:3.2.5:test (default-test) @ Mu
run.
ecution of surefire because it h
SS
1.929 s
: 2025-09-02T11:19:11+05:30

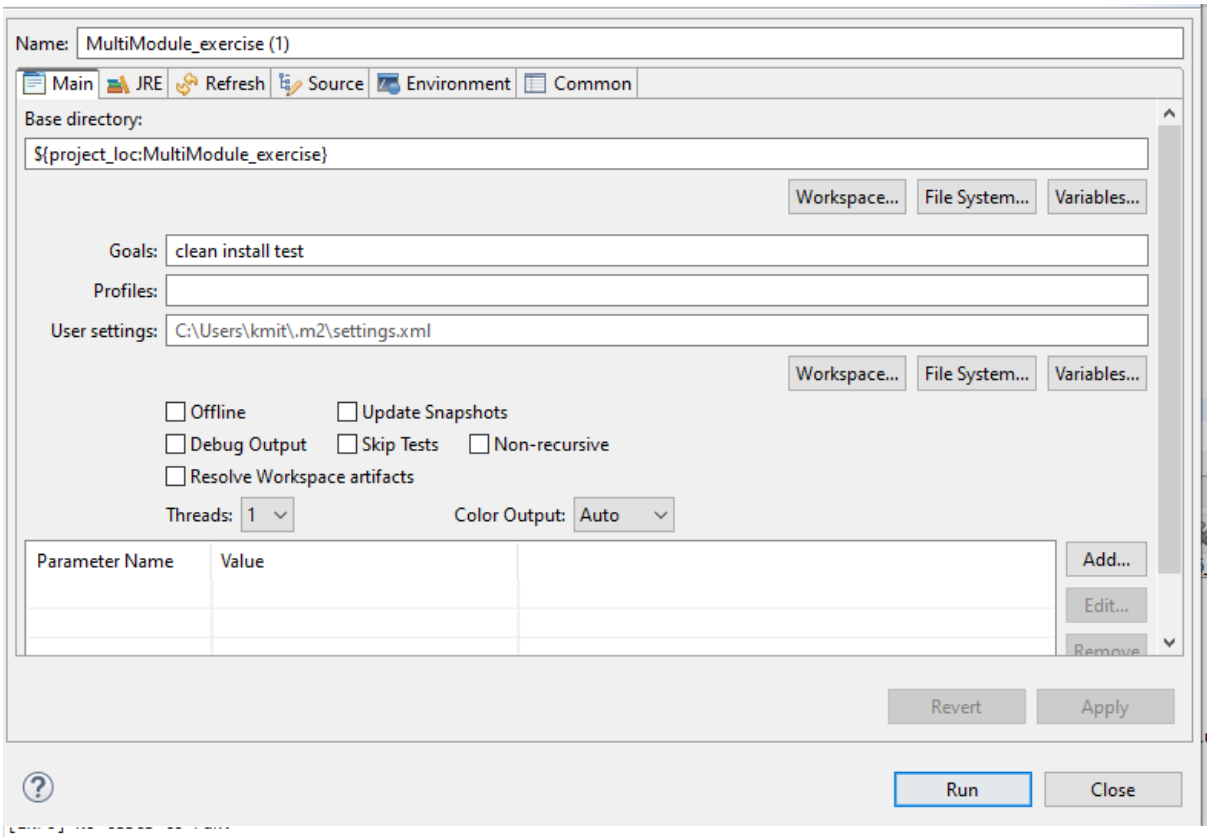

New Maven Module
New Maven Module
Select the module name and parent
Create a simple project (skip archetype selection)
Module Name: MultiModuleChild1_week-7
Parent Project: MultiModule_exercise
Browse...
Add project(s) to working set
Working set:
More...
Advanced
< Back
Next >
Finish
Cancel


New Maven Module
New Maven Module
Select the module name and parent
Create a simple project (skip archetype selection)
Module Name:
MultiModuleChild2_week-7
Parent Project:
MultiModule_exercise
Browse...
Add project(s) to working set
Working set:
More...
Advanced
< Back
Next >
Finish
Cancel



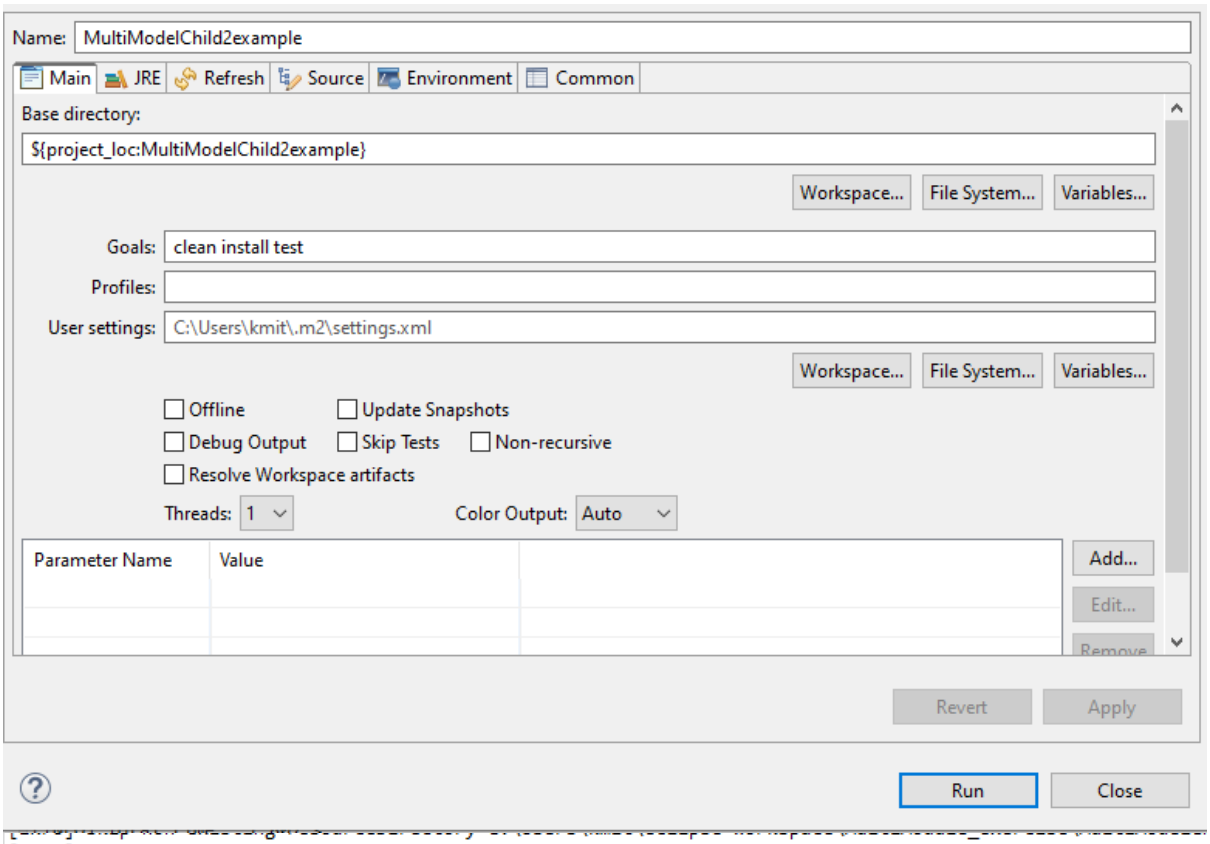




```
<terminated> C:\Users\kmit\Documents\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe (02-Sept-2025, 12:26:42
[INFO] Parameter: package, Value: org.MultiModelChild2example
[INFO] Parameter: groupId, Value: KMIT-WEEK-7
[INFO] Parameter: artifactId, Value: MultiModelChild2example
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[WARNING] CP Don't override file C:\Users\kmit\eclipse-workspace\MultiModule_exercise\MultiModelChild2example\src\main\webapp
[WARNING] CP Don't override file C:\Users\kmit\eclipse-workspace\MultiModule_exercise\MultiModelChild2example\.mvn
[INFO] Project created from Archetype in dir: C:\Users\kmit\eclipse-workspace\MultiModule_exercise\MultiModelChild2example
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  40.094 s
[INFO] Finished at: 2025-09-02T12:27:28+05:30
[INFO] ------------------------------------------------------------------------
```




```
[INFO] Skipping execution of surefire because it has already been run for this configuration
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for MultiModule_exercise 0.0.1-SNAPSHOT:
[INFO]
[INFO] MultiModule_exercise ............................... SUCCESS [  0.498 s]
[INFO] MultiModuleChild1_week-7 ........................... SUCCESS [  4.735 s]
[INFO] MultiModelChild2example Maven Webapp ............... SUCCESS [  3.408 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.800 s
[INFO] Finished at: 2025-09-02T12:29:30+05:30
[INFO] ------------------------------------------------------------------------
```

```
[INFO] Copying 0 resource from src\test\resources to target\test-classes
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ MultiModuleChild1_week-7 ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ MultiModuleChild1_week-7 ---
[INFO] Skipping execution of surefire because it has already been run for this configuration
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.807 s
[INFO] Finished at: 2025-09-02T12:29:59+05:30
[INFO] ------------------------------------------------------------------------
```




```
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ MultiModelChild2example ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.3.0:test (default-test) @ MultiModelChild2example ---
[INFO] No tests to run.
[INFO] Skipping execution of surefire because it has already been run for this configuration
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.088 s
[INFO] Finished at: 2025-09-02T12:30:30+05:30
[INFO] ------------------------------------------------------------------------
```

# SBQs(week 8)

**1. Q:** What is Jenkins primarily used for?
**A:** Jenkins is used for Continuous Integration and Continuous Delivery (CI/CD) automation.

**2. Q:** What is a feature of Jenkins?
**A:** Automates building, testing, and deploying code using pipelines and plugins.

**3. Q:** What is the default port on which Jenkins runs?
**A:** Port **8080**.

**4. Q:** What can be integrated with Jenkins for version control?
**A: Git**, **GitHub**, **Bitbucket**, **GitLab**, and **SVN**.

**5. Q:** What is the purpose of Jenkins plugins?
**A:** To extend Jenkins functionality and integrate external tools.

**6. Q:** Which type of Jenkins job is best for one-off tasks or small scripts?
**A: Freestyle Project**.

**7. Q:** How can you manage sensitive information such as API keys in Jenkins?
**A:** Use Credentials Manager to store and access them securely.

**8. Q:** What does the "Blue Ocean" feature in Jenkins refer to?
**A:** A modern, user-friendly UI for visualizing and managing pipelines.

**9. Q:** What does the "Blue Ocean" feature in Jenkins refer to?
**A:** It provides a graphical view of pipeline stages and builds.

**10. Q:** Which Jenkins component allows distributed builds across multiple machines?
**A:** Jenkins agents (nodes)

**11. Q:** List five important Jenkins plugins for microservices CI/CD and their purposes.
**A:** Git Plugin – Integrates Git repositories.

Pipeline Plugin – Enables CI/CD pipeline scripting.

Docker Plugin – Builds and deploys Docker containers.

Kubernetes Plugin – Runs builds in Kubernetes pods.

JUnit Plugin – Publishes and visualizes test results.

12. **Q:** Steps to install a plugin via Jenkins UI and key considerations?
**A:** Go to Manage Jenkins → Manage Plugins → Available.

Search and install the plugin.

Restart Jenkins.

Check compatibility with Jenkins version and dependencies before installing.

13. **Q:** Steps to install a plugin via Jenkins UI and key considerations?
A:Navigate to Manage Jenkins → Manage Plugins → Available tab.

Install selected plugin and restart Jenkins.

Verify update logs and ensure version compatibility.

14. Q: After installing the Git Plugin, how to configure it?

A:Go to Manage Jenkins → Global Tool Configuration.

Add Git installation path.

In pipeline, use:

```
git url: 'https://github.com/user/repo.git', credentialsId: 'git-creds'.
```

15. **Q:** Common plugin issues and how to troubleshoot them?
**A: Issues:** Dependency conflicts, version mismatches, plugin crashes.

**Fixes:** Check Jenkins logs, update plugins, verify compatibility matrix, or rollback versions.

**Week 9: Pipeline Creation using script**

1. Evaluation of Jenkins pipeline.
2. **WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE**
3. Hands-on practice on creation of scripted Jenkins pipeline.
4. Take the screenshots for above task

Procedure

**General :**

Description :- provide the description of the project

Build triggers: here we can provide with build triggers of you choice

**Advance project option**:

Definition:

Choose pipeline script

**Here code for pipeline -script**

Copy the code there

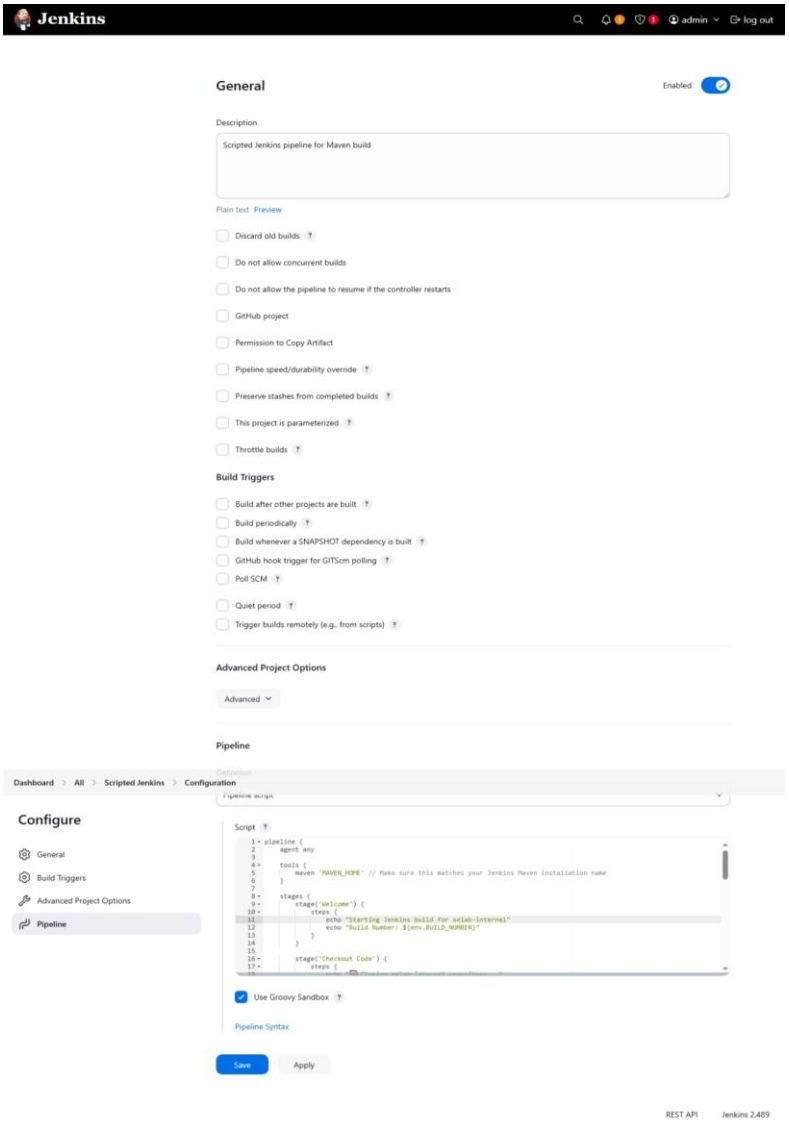
---

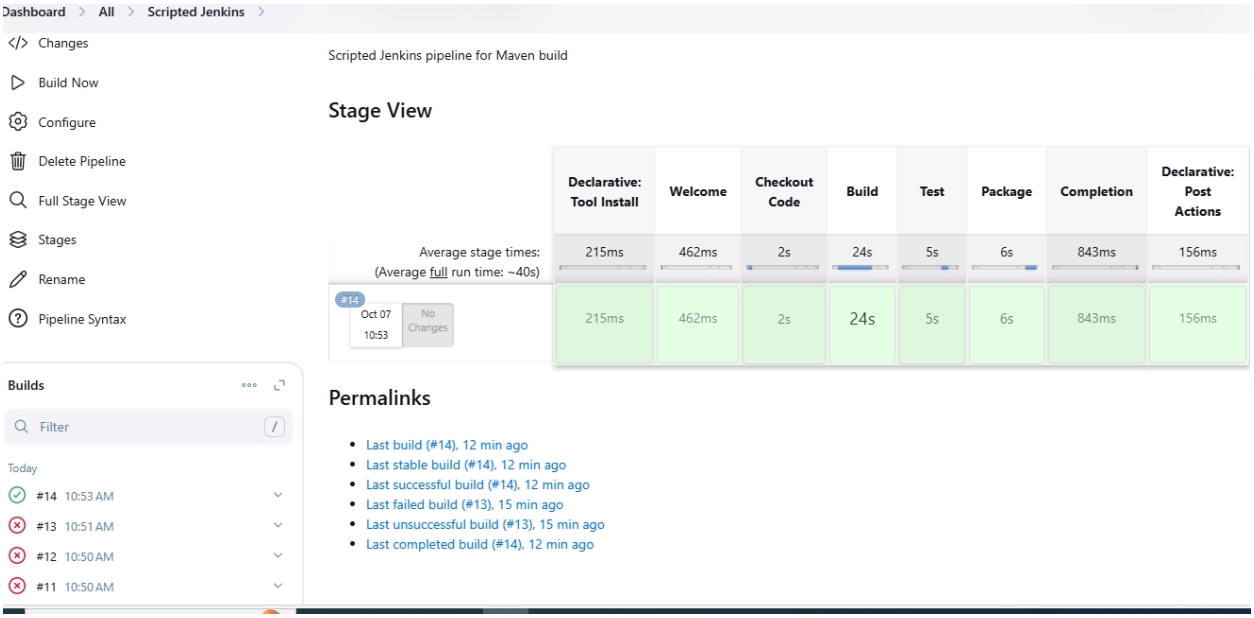
```
pipeline {
   agent any
   tools{
      maven 'MAVEN-HOME'
   }
   stages {
      stage('git repo & clean') {
         steps {
            //bat "rmdir  /s /q mavenjava"
            bat "git clone provide your github link"
            bat "mvn clean -f mavenjava"
         }
      }
      stage('install')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

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

---

Apply and save


Jenkins
admin
log out
General
Enabled
Description
Scripted Jenkins pipeline for Maven build
Plain text Preview
Discard old builds
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Permission to Copy Artifact
Pipeline speed/durability override
Preserve stashes from completed builds
This project is parameterized
Throttle builds
Build Triggers
Build after other projects are built
Build periodically
Build whenever a SNAPSHOT dependency is built
GitHub hook trigger for GITScm polling
Poll SCM
Quiet period
Trigger builds remotely (e.g., from scripts)
Advanced Project Options
Advanced
Pipeline
Definition
Dashboard > All > Scripted Jenkins > Configuration
Configure
General
Build Triggers
Advanced Project Options
Pipeline
Script
pipeline {
agent any
tools {
maven 'MAVEN_HOME' // Make sure this matches your Jenkins Maven installation name
}
stages {
stage('Welcome') {
steps {
echo "Starting Jenkins build for selab-internal"
echo "Build Number: ${env.BUILD_NUMBER}"
}
}
stage("Checkout Code") {
steps {
Use Groovy Sandbox
Pipeline Syntax
Save
Apply
REST API
Jenkins 2.489




SBQ's:

1. **Your manager asks you to clean the workspace before building — which stage in this pipeline takes care of it?**

**Ans:** The Build stage (inside the dir('selab-internal') block) runs mvn clean install, and additionally, the Checkout Code stage removes any existing folder with rmdir /s /q selab-internal, which ensures the workspace is cleaned before the build.

2. **The GitHub repository link is missing in the script — where exactly do you provide it?**

**Ans:** The GitHub link is provided in the Checkout Code stage in the line:

git clone https://github.com/NaveenCK-10/selab-internal.git

3. **If Maven is not configured globally in Jenkins, which section of the pipeline will fail first?**

**Ans:** he pipeline will fail at the Build stage where Maven is invoked using:

mvn clean install

because Jenkins won't be able to locate Maven without a global tool configuration.

4. **A teammate complains the pipeline is not creating .war files — which stage is responsible?**

**Ans:** The Package stage is responsible for creating the final .jar or .war artifacts using:

mvn package

5. **Your test cases failed, but the pipeline still continued — how will Jenkins behave in the test stage?**

**Ans: By default, if a Maven test fails, it throws an error and the pipeline stops. In our current script, the pipeline stops at the Test stage on failure, and subsequent stages (Package) are skipped.**

6. **Instead of running nightly builds, you want this pipeline to trigger only when GitHub changes occur — where will you configure it?**

**Ans:** You configure it in Build Triggers → GitHub hook trigger for GITScm polling in Jenkins. Alternatively, a GitHub webhook can notify Jenkins to trigger the pipeline on every push.

7. **If you replace mvn clean with mvn compile, what difference will it make to the project build?**

**Ans:** mvn clean deletes previous build artifacts to avoid conflicts, while mvn compile only compiles the source code without cleaning previous outputs. Replacing clean with compile may lead to using old artifacts in the build.

8. **The project folder name is not mavenjava but studentapp — which parts of the script must you edit?**

**Ans:** Update all references of dir('selab-internal') to dir('studentapp') and also the rmdir /s /q and git clone folder name accordingly.

9. **If the install stage fails, will the test and package stages still run in this pipeline?**

**Ans:** No. In a declarative Jenkins pipeline, if a stage fails (here Build or Install), the following stages (Test and Package) are skipped automatically.

10. **A student asks where to add deployment steps to Tomcat after packaging — which is the best place in this script?**

**Ans:** Deployment steps should be added after the Package stage, ideally in a new stage called Deploy that runs commands like copying .war to Tomcat's webapps directory.

11. **Why is tools { maven 'MAVEN-HOME' } used in this pipeline?**

**Ans: It tells Jenkins which Maven installation to use so that Maven commands like mvn clean install can run properly.**

12. **If GitHub credentials are private, how can you secure them in the git repo & clean stage?**

**Ans:** We can store GitHub credentials in Jenkins Credentials and reference them using credentialsId in the git command:

git branch: 'main', credentialsId: 'github-credentials-id', url: 'https://github.com/...'

13. **Which Jenkins plugin is needed to recognize the pipeline { ... } structure in this script?**

**Ans:** We need the Pipeline plugin (also called Workflow plugin) to understand the pipeline { ... } structure.

14. **In Windows, the script uses bat commands — what change would you make if Jenkins runs on Linux?**

Ans: If Jenkins runs on Linux, we should replace all `bat` commands with `sh` commands.

For example: `'mvn clean install'`

15. **How will you modify the pipeline to stop execution if the GitHub clone command fails?**

**Your project lead asks you to print a welcome message in Jenkins to confirm the pipeline is working — how would you script it?**

**Ans:** We can add || exit 1 at the end of the git clone command so Jenkins stops immediately if cloning fails. Or we can use checkout step which fails the stage automatically on error.

16. **A teammate forgot to pull the latest GitHub code before building; how can you fix this in your pipeline?**

**Ans:** We can add an initial stage like this:

```
stage('Welcome') {
  steps {
    echo "Hello! Jenkins pipeline is running for selab-internal"
  }
}
```

It shows a friendly message in the console when the pipeline starts.

17. **Your Java file Hello.java must be compiled every time code is committed — how will you add this step?**

**Ans:** We can include mvn compile or mvn install in the Build stage. Since Maven automatically compiles Java files, every commit will trigger compilation when the pipeline runs.

18. **Tests should stop the pipeline if they fail — where will you put the mvn test command?**

**Ans:** Keep mvn test in the Test stage. By default, if any test fails, Maven returns an error, and Jenkins stops the pipeline automatically.

19. **Every evening at 6 PM your pipeline should run automatically — how can you set this in Jenkins?**

**Ans:** We can add a cron schedule in Jenkins Build Triggers. For example:

0 18 * * *

This runs the pipeline at 6:00 PM daily.

20. **You only want to package the project if compilation succeeds — how would you connect the stages?**

**Ans:** Jenkins declarative pipelines automatically stop stages on failure. So if Build (compile/install) fails, the Package stage will not run. This ensures packaging happens only after successful compilation.

21. **Your professor wants a .jar file generated for submission — what pipeline stage will you add?**

**Ans:** The .jar file is created in the Package stage using:

mvn package

It will be located in selab-internal/target/ after a successful build

22. **A Git clone step is failing due to wrong credentials — how would you secure and use them in your script?**

**Ans:** We should store GitHub credentials in Jenkins Credentials and use them in the pipeline:

git branch: 'main', credentialsId: 'github-credentials-id', url: 'https://github.com/NaveenCK-10/selab-internal.git'

This keeps credentials secure and allows Jenkins to clone private repos.

23. **A teammate wants to see the build number inside console logs — how do you print it in the pipeline?**

**Ans:** We can print the build number using the environment variable ${env.BUILD_NUMBER}. For example:

echo "🔢 Build Number: ${env.BUILD_NUMBER}"

It will appear in the console every time the pipeline runs.

# LAB ACTION PLAN FOR WEEK 10

## Working with minikube and Nagios

1. Hands-on practice of creating, running and scaling pods in minikube.
2. Running Nginx server on specified port number by explaining the Nginx monitoring tool
3. Running Nagios server and Understanding the Monitoring tool using Docker.
4. AWS-free Trier account Creation steps
5. Upload the screenshots for the tasks

**Kubernetes:**

Kubernetes is a tool that automates how we run and manage applications inside the container.

Dockers will only run containers, if in any case the container fails/stopped/killed, the docker will not help us, here is where Kubernetes plays an important role, Kubernetes cluster will be responsible in creating a new container and managing various containers.

**POD:** In Kubernetes, a Pod is the smallest deployable unit that you can create and manage.

**Minikube:**

Minikube creates a VM on your local machine and deploys a simple Kubernetes cluster with one node. It's a lightweight implementation.Minikube *is* a version of Kubernetes.

**Nagios:**

Nagios is an **open-source IT infrastructure monitoring tool**. It monitors

- Serrvers
- Network devices
- Applications and services

It **alerts admnistrators** when issues occur and notifies when they are resolved.

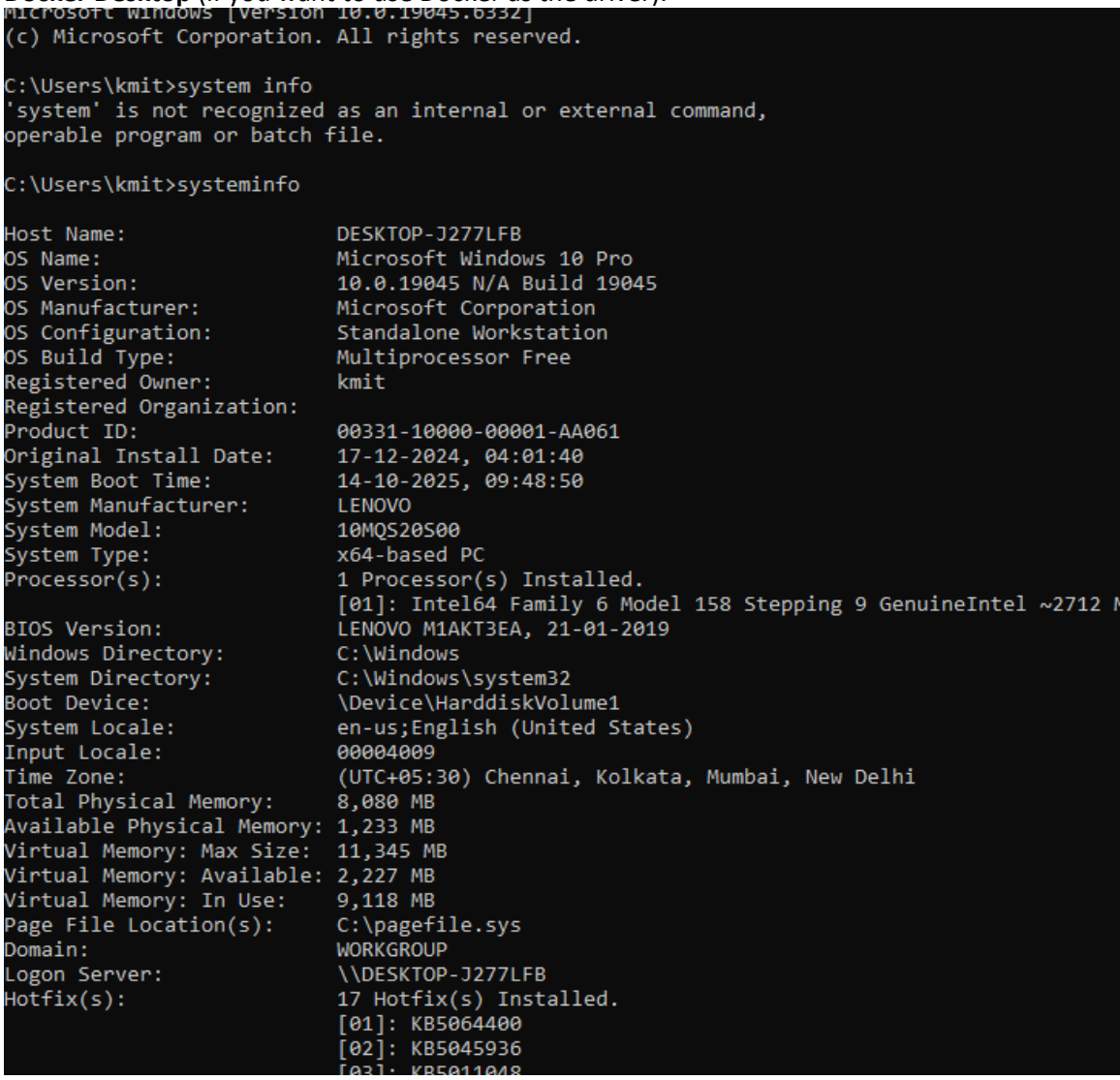
---

Step 1: Install Prerequisites

Before installing Minikube, ensure the following are installed:

1. **Virtualization Support**:
   - Verify virtualization is enabled:

2. **Hypervisor**:
    - Minikube supports multiple hypervisors (e.g., **Hyper-V**, **VirtualBox**, or **Docker** as a driver).
    - Install one of the following:
        - **Hyper-V** (pre-installed on Windows 10/11 Pro or Enterprise).
        - **Docker Desktop** (if you want to use Docker as the driver).




---

Step 2: Download Minikube

1. Open a PowerShell or Command Prompt with administrator privileges.
2. Download the latest Minikube executable using this command:
3. `curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe`
4. Install Minikube by running the installer:
5. `.\minikube-installer.exe`

---

Step 3: Add Minikube to PATH

If Minikube is not automatically added to your PATH during installation:

1. Open **System Properties → Environment Variables**.
2. Add the directory where Minikube is installed (e.g., `C:\Program Files\Minikube`) to your PATH variable.

---

Step 4: Start Minikube

1. Open a terminal (PowerShell or CMD). Do the following commands
2. Start Minikube with a specified driver (e.g., Hyper-V, Docker, or VirtualBox). For example:
    - **Hyper-V**:
    - `minikube start --driver=hyperv`
    - **Docker**:
    - `minikube start --driver=docker`
3. Verify Minikube is running:
4. `minikube status`

```
C:\Users\kmit>minikube start --driver=docker
* minikube v1.34.0 on Microsoft Windows 10 Pro 10.0.19045.6332 Build 19045.6332
* minikube 1.37.0 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.37.0
* To disable this notice, run: 'minikube config set WantUpdateNotification false'

* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\kmit>
C:\Users\kmit>kubectl get nodes
NAME       STATUS   ROLES           AGE    VERSION
minikube   Ready    control-plane   293d   v1.31.0

C:\Users\kmit>
```

---

Step 5: Interact with Minikube

kubectl is a command-line tool used in Kubernetes to interact with and manage Kubernetes clusters.

Once Minikube is running:

1. Use `kubectl` to interact with the cluster.

- Install `kubectl` if not already installed:
- `minikube kubectl -- get pods -A`
- Or download it separately from the official Kubernetes site.

2. Open the Minikube dashboard (optional):
3. `minikube dashboard`

```
C:\Users\kmit>kubectl get pods -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS        AGE
kube-system   coredns-6f6b679f8f-8wfqf           1/1     Running   1 (5m12s ago)   293d
kube-system   coredns-6f6b679f8f-ptm4v           1/1     Running   1 (5m12s ago)   293d
kube-system   etcd-minikube                      1/1     Running   1 (5m12s ago)   293d
kube-system   kube-apiserver-minikube            1/1     Running   1 (5m12s ago)   293d
kube-system   kube-controller-manager-minikube   1/1     Running   1 (5m12s ago)   293d
kube-system   kube-proxy-wxt5w                   1/1     Running   1 (5m12s ago)   293d
kube-system   kube-scheduler-minikube            1/1     Running   1 (5m12s ago)   293d
kube-system   storage-provisioner                1/1     Running   3 (4m38s ago)   293d

C:\Users\kmit>
```

---

Optional: Check Your Installation

**Run the following to verify the installation:**

Minikube version

`kubectl version --client`

```
C:\Users\kmit>kubectl version --client
Client Version: v1.30.5
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3

C:\Users\kmit>
```

---

**Troubleshooting**

1. **If Minikube fails to start**:
   - Ensure your hypervisor (Hyper-V/Docker/VirtualBox) is installed and running.
   - Check the Minikube logs:
   - `minikube logs`
2. **Updating Minikube**:
3. `minikube update-check`
4. `minikube update`

## Minikube Automation Steps

### Step 1: Start Minikube Cluster

- Open your terminal and run the command:

```
minikube                                                    start
```

```
C:\Users\kmit>minikube start
* minikube v1.34.0 on Microsoft Windows 10 Pro 10.0.19045.6332 Build 19045.6332
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Updating the running docker "minikube" container ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\kmit>
```

**Step 2: Create and Manage Deployment**

1. **Create an application in Kubernetes:**

   - Command:

     ```
     kubectl create deployment mynginx--image=nginx
     ```

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

     C:\Users\kmit>
     ```

     ```
     if already created then
     ```

     ```
     kubectl set image deployment/myngnix nginx=nginx:latest
     ```

   - Verify the deployment using: Kubernetes responds by showing you a list that includes the names of your deployment groups

     ```
     kubectl get deployments
     ```

     ```
     C:\Users\kmit>kubectl get deployments
     NAME      READY   UP-TO-DATE   AVAILABLE   AGE
     mynginx   1/1     1            1           57s

     C:\Users\kmit>
     ```

- Ensure `mynginx` appears in the output.

Check the following commands:

- kubectl get pods

```
C:\Users\kmit>kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
mynginx-79bb8756c7-xm6g5     1/1     Running   0          92s

C:\Users\kmit>
```

- kubectl describe pods

```
C:\Users\kmit>kubectl describe pods
Name:             mynginx-79bb8756c7-xm6g5
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Tue, 14 Oct 2025 10:23:08 +0530
Labels:           app=mynginx
                  pod-template-hash=79bb8756c7
Annotations:      <none>
Status:           Running
IP:               10.244.0.8
IPs:
  IP:           10.244.0.8
Controlled By:  ReplicaSet/mynginx-79bb8756c7
Containers:
  nginx:
    Container ID:   docker://c8378173f47afd0611077a51b228cf13893130cbe4ffa08accd06a08799a7455
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Tue, 14 Oct 2025 10:23:36 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-h6lhm (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-h6lhm:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
```

2. **Expose Deployment as a Service**:
    - Command:

```
kubectl expose deployment mynginx --type=NodePort --port=80 --
target-port=80
```

```
C:\Users\kmit>kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80
service/mynginx exposed

C:\Users\kmit>
```

**Step 3: Scale the Deployment**

Command:Scales the Nginx deployment to 4 replicas (pods).

```
kubectl scale deployment mynginx --replicas=4
```

```
C:\Users\kmit>kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled

C:\Users\kmit>
```

```
kubectl get service myngnix
```

```
C:\Users\kmit>kubectl get service mynginx
NAME      TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
mynginx   NodePort   10.102.91.131   <none>        80:32151/TCP   3m28s

C:\Users\kmit>
```

**Step 4: Access the Nginx App**

1. **Using Port Forwarding**:
    - Command:

```
kubectl port-forward svc/mynginx 8081:80
```

```
C:\Users\kmit>kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

- Access the app via http://localhost:8081.


If Error, use this option, **Using Minikube Tunnel**:

- Start the tunnel:

  ```
  minikube tunnel
  ```

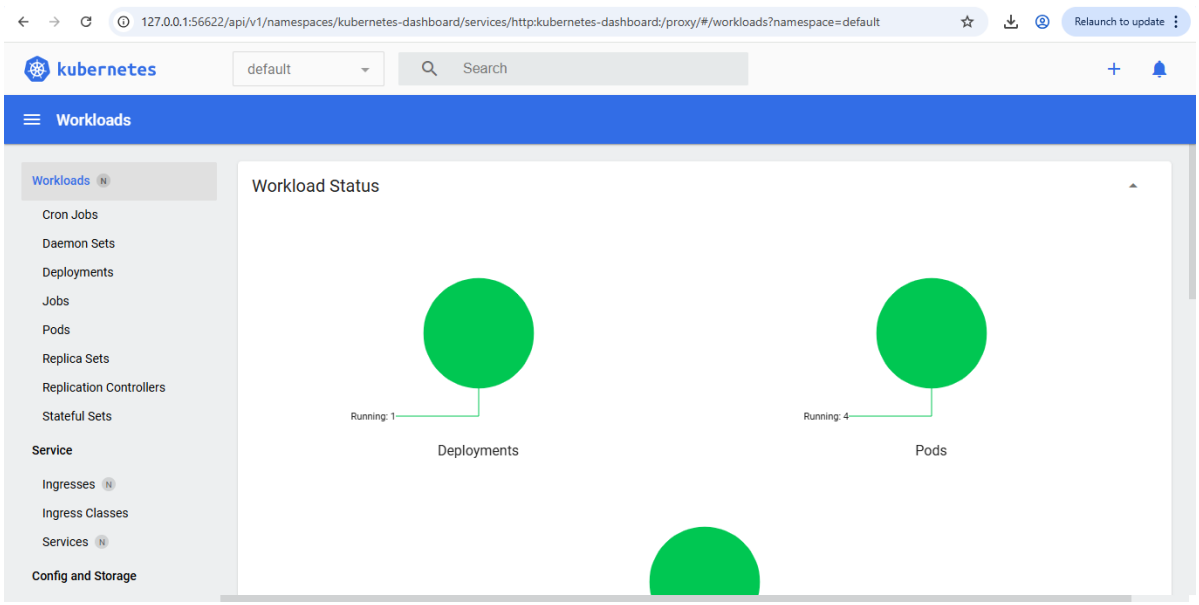
- Retrieve the service URL:

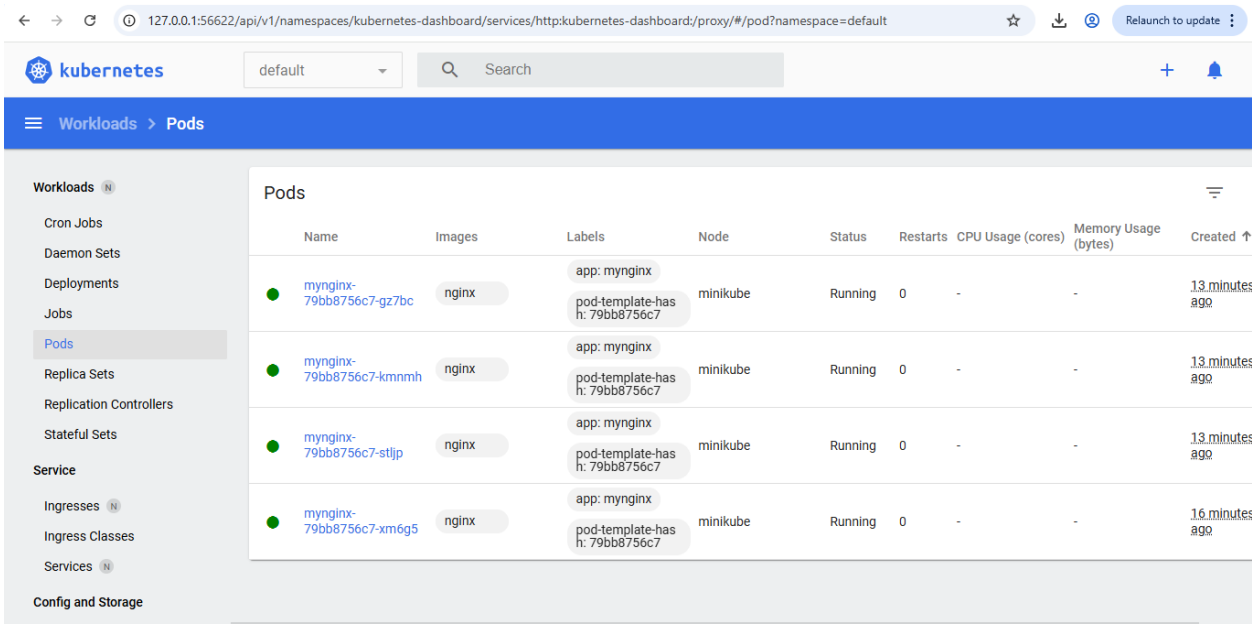
  ```
  minikube service mynginx --url
  ```

- Open the provided URL in your browser.
- Open the kubernets dashboard

- Open the minikube dashboard

  Minikube dashboard




## Step 5: Stop and Clean Up

1. **Stop Nginx Deployment**:
    - Commands:

```
kubectl delete deployment mynginx
```

```
C:\Users\kmit>kubectl delete deployment mynginx
deployment.apps "mynginx" deleted

C:\Users\kmit>
```

```
kubectl delete service mynginx
```

```
C:\Users\kmit>kubectl delete service mynginx
service "mynginx" deleted

C:\Users\kmit>
```

2. **Stop Minikube (Optional)**:
    - Command:

```
minikube stop
```

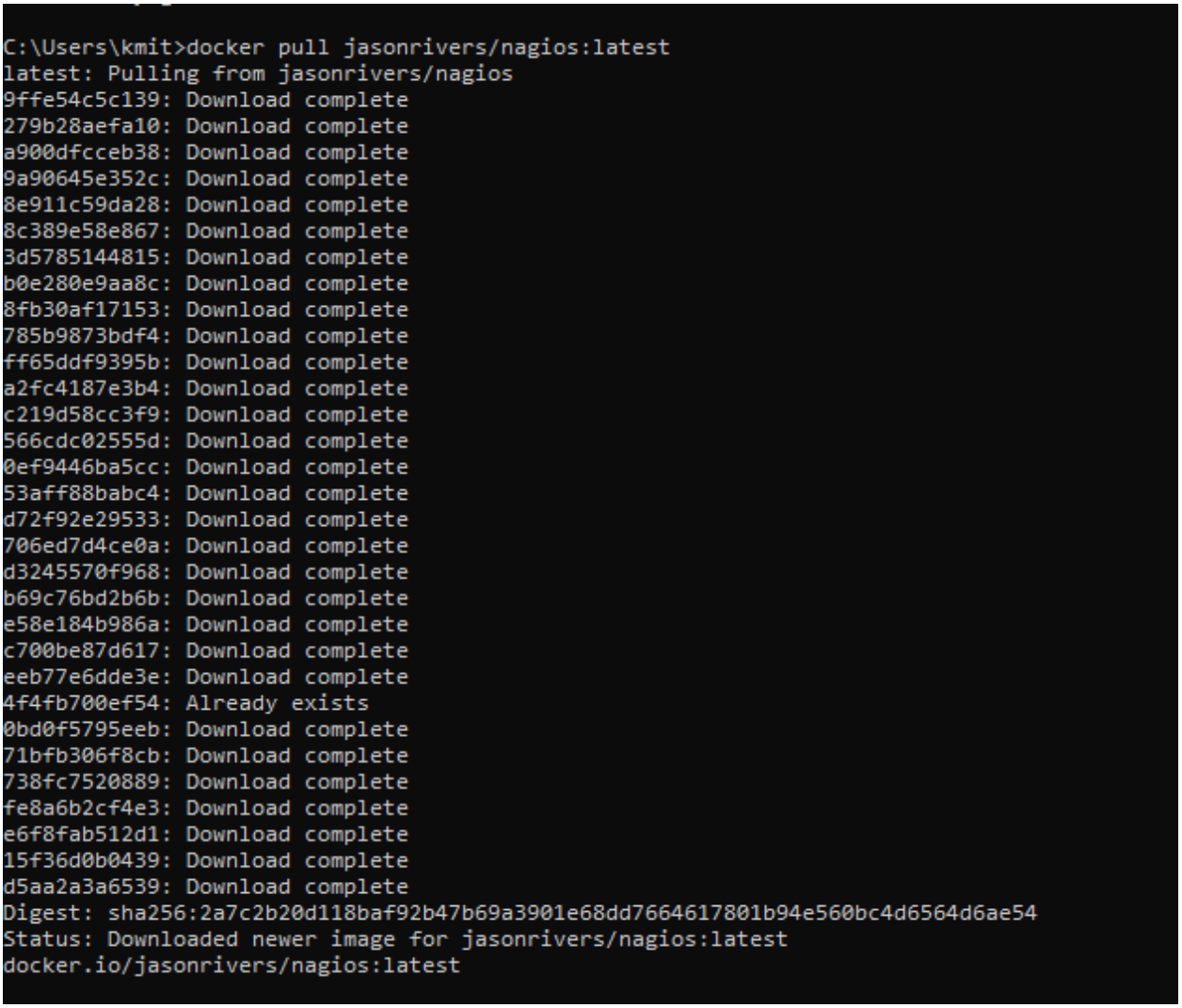
3. **Delete Minikube Cluster (Optional)**:
    - Command:`minikube delete`

## Nagios Automation Steps

### Step 1: Pull the Nagios Docker Image

- Open a terminal and run:

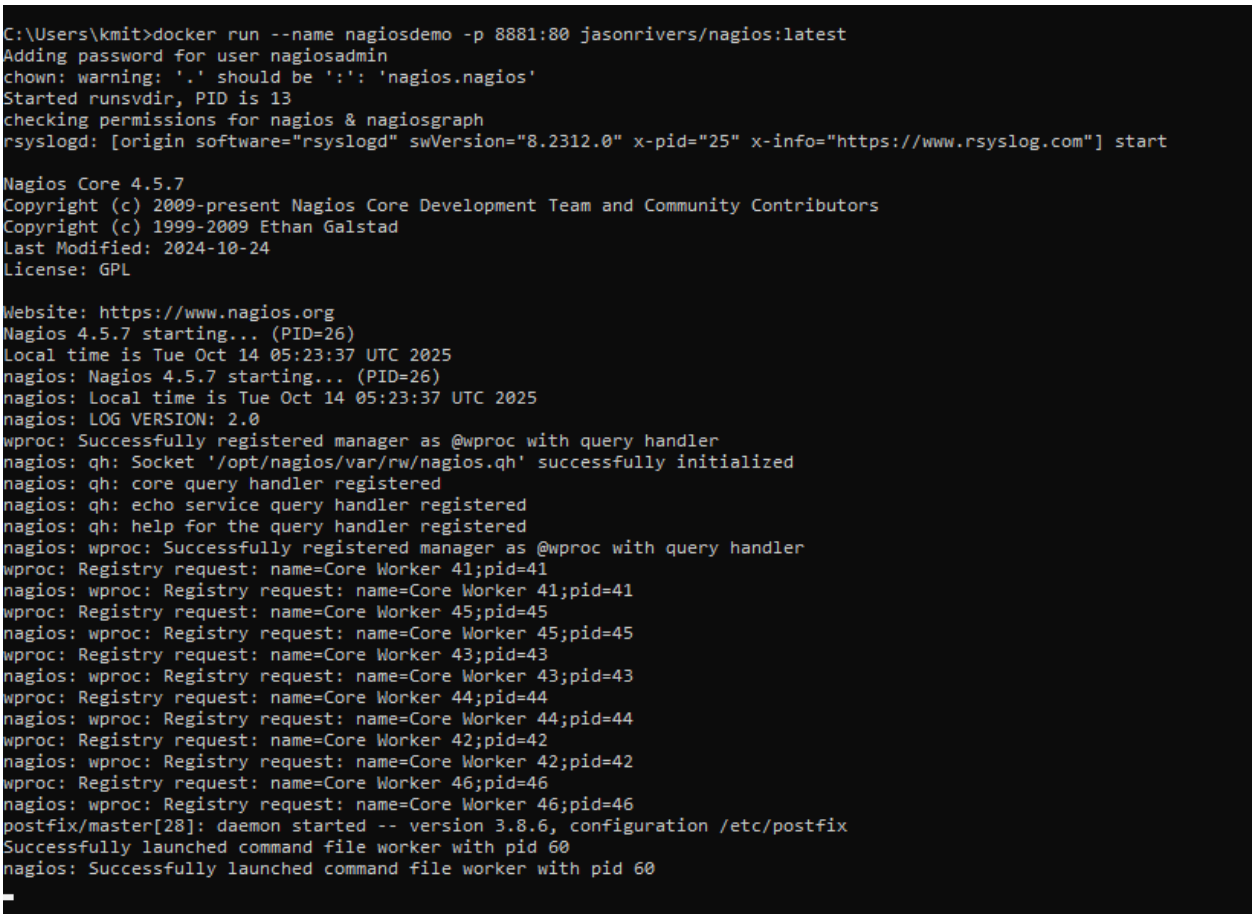
```
docker pull jasonrivers/nagios:latest
```




**Step 2: Run Nagios**

- Command:

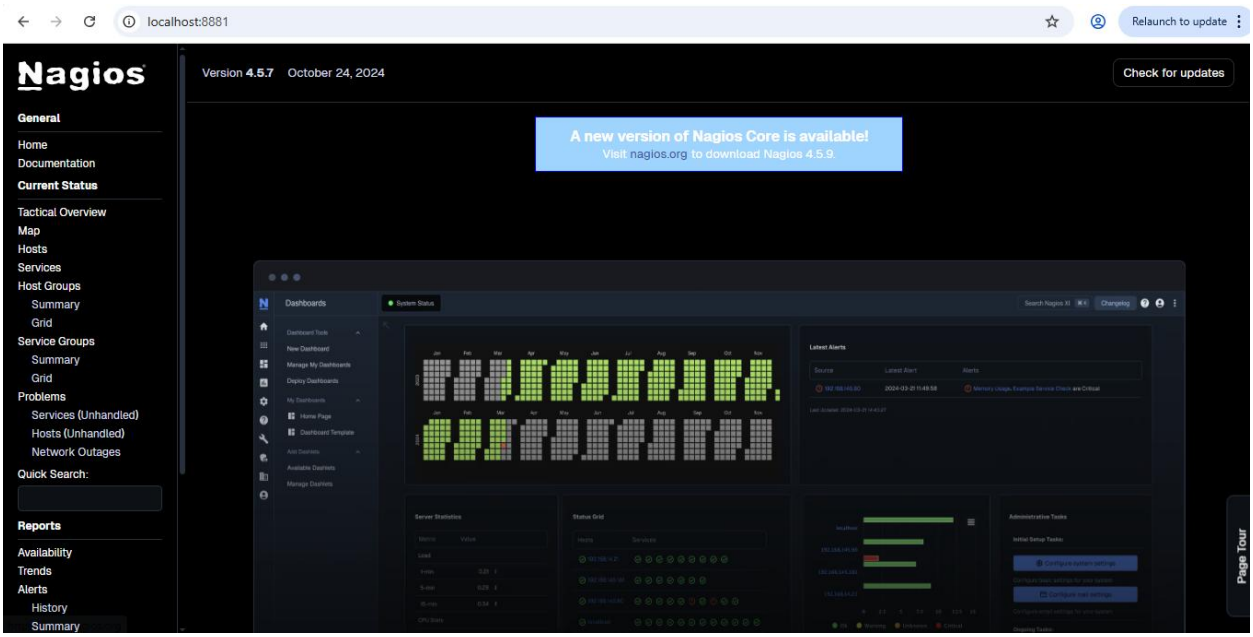
```
docker run --name nagiosdemo -p 8888:80 jasonrivers/nagios:latest
```



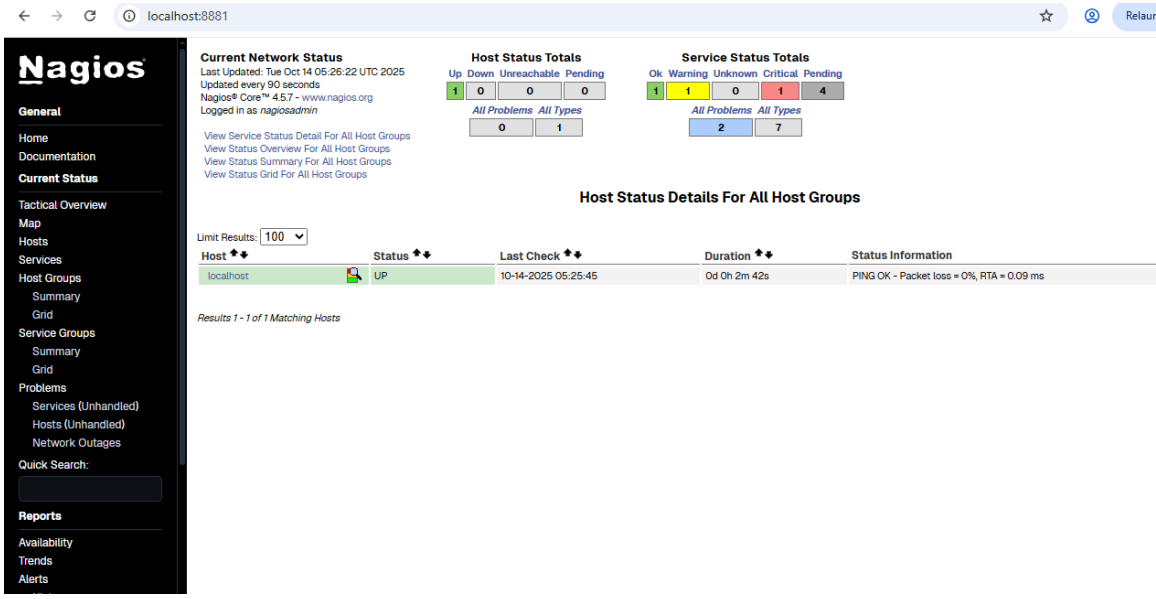

### Step 3: Access Nagios Dashboard

- Open your browser and navigate to:

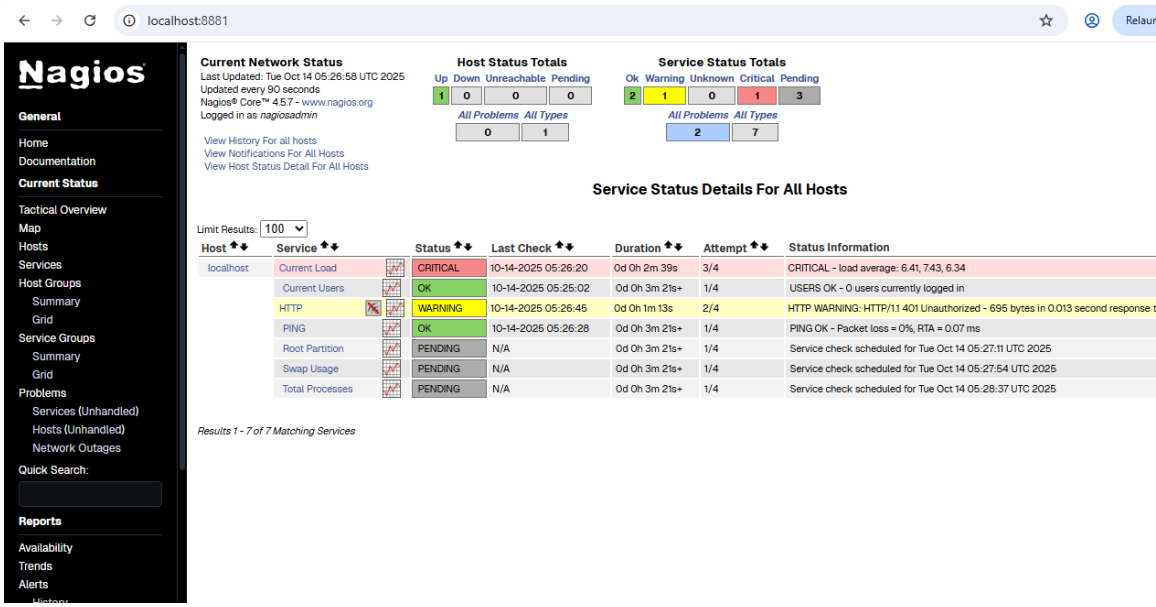
  http://localhost:8888



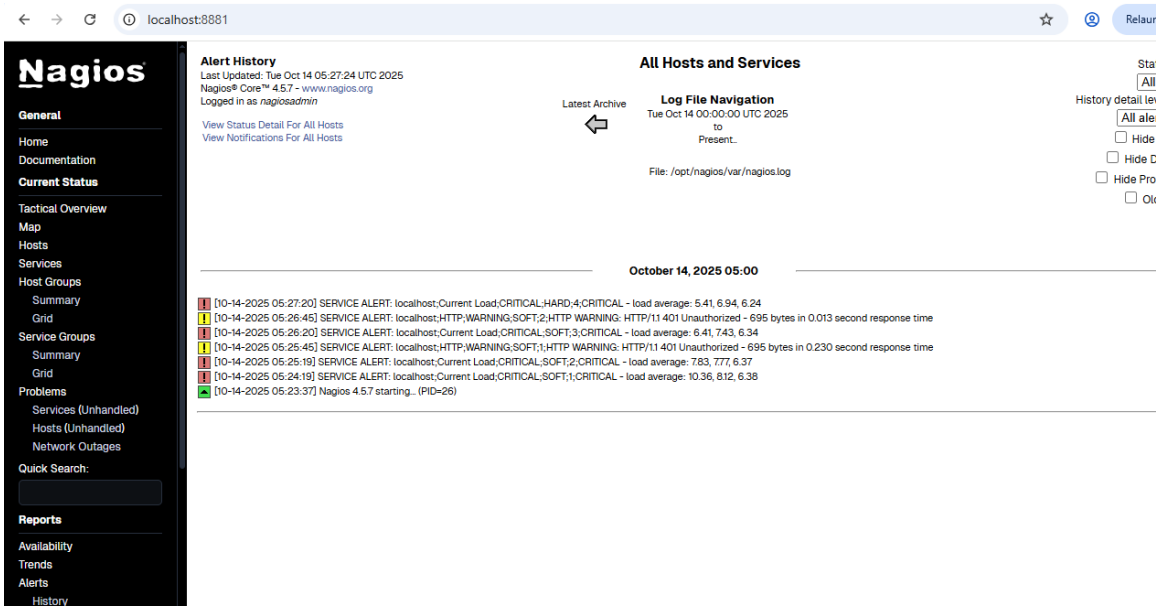

- **Login Credentials**:
  - Username: `nagiosadmin`
  - Password: `nagios`
- Once logged in, explore:
  - Hosts: View systems being monitored.




  - Services: Check tasks being monitored (e.g., CPU usage).



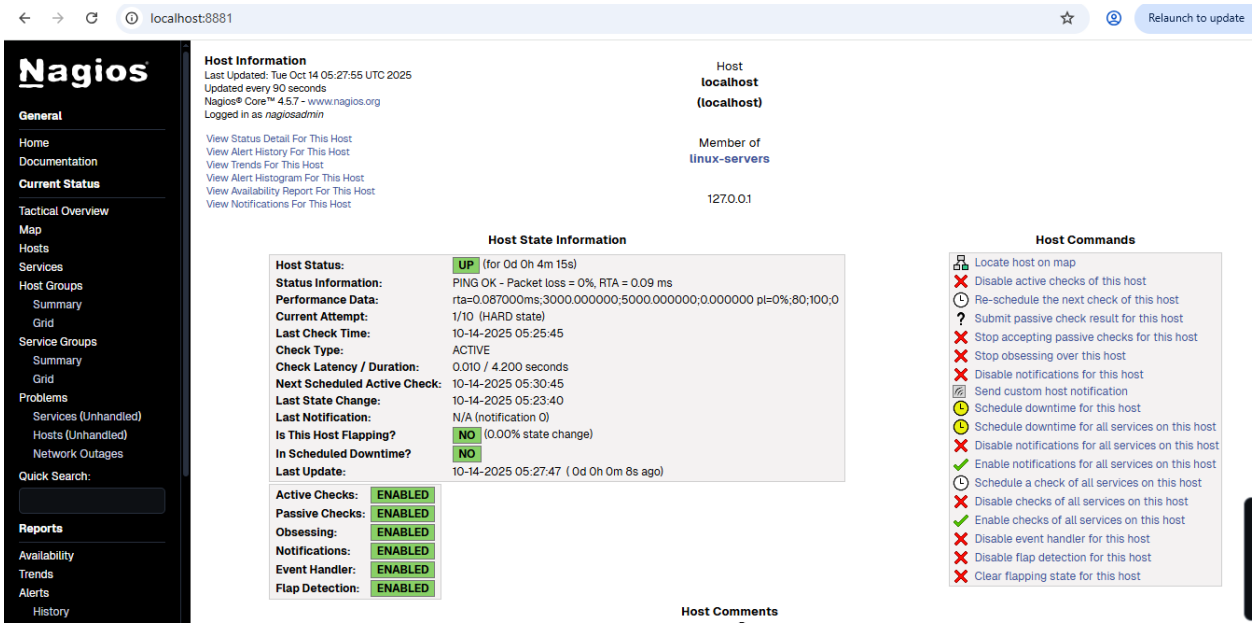

- Alerts: Access recent notifications.




## Step 4: Monitoring Host Details

1. **Navigate to the Host Information Page**:
   - Select a host from the **Hosts** menu.
2. **Key Details**:
   - Host Status: Indicates if the system is UP or DOWN.

- Metrics: View CPU usage, memory status, and network activity.
- Actions: Reschedule checks, disable notifications, or schedule downtime.




## Step 5: Stop and Remove Nagios

1. **Stop the Container**:
   - Command:

     ```
     docker stop nagiosdemo
     ```

     ```
     C:\Users\kmit>
     C:\Users\kmit>docker stop nagiosdemo
     nagiosdemo

     C:\Users\kmit>
     ```

2. **Delete the Container**:
   - Command:

     ```
     docker rm nagiosdemo
     ```

```
C:\Users\kmit>docker rm nagiosdemo
nagiosdemo

C:\Users\kmit>
```

3. **Remove the Image** (Optional):
    - List images:

      ```
      docker images
      ```

    - Delete the Nagios image:

      ```
      docker rmijasonrivers/nagios:latest
      ```

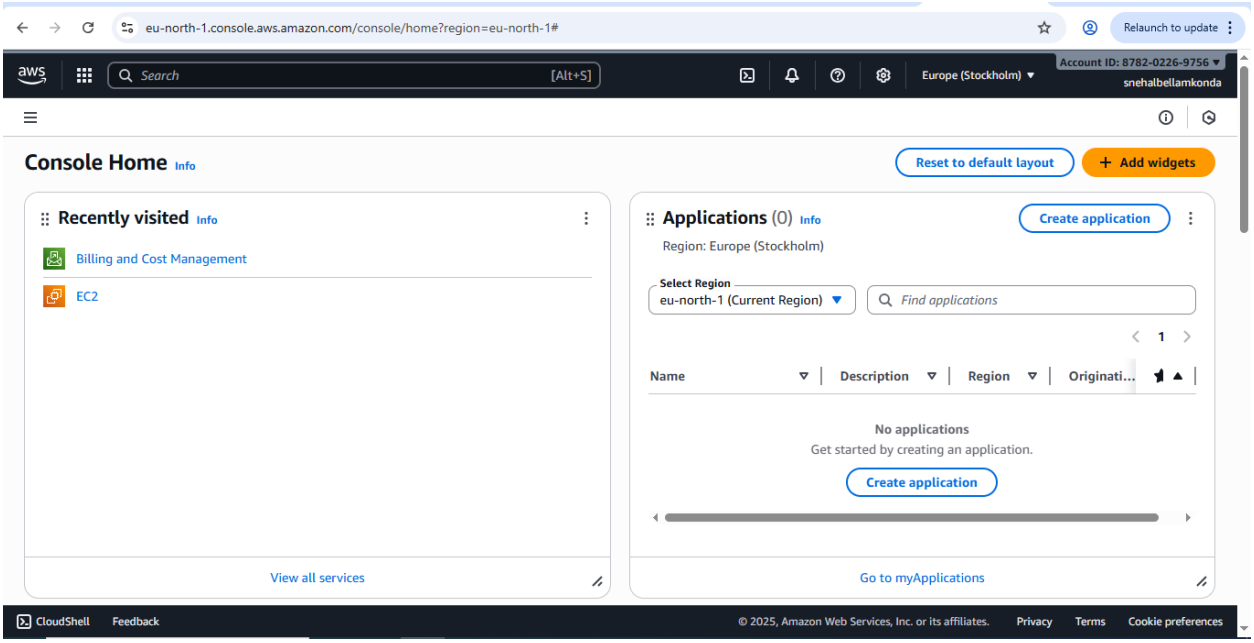
4. Observe the docker containers in DockerHub, we can see the latest Nagios Installed running on port:8888

---

**Steps for AWS-free trier account creation**

- open https://aws.amazon.com
- click on create AWS account
- Provide email id and name details for AWS account creation then a screen will appear as below
- Enter verification code received in your mail
- Provide your contact details and agree Terms and conditions, then click on create
- Provide billing details, click on verify and continue
- Amazon chagres 2 rs and it will be credit back into your account once verification is over
- Once payment success a screen appear as below, provide your working contact number after selecting country, and click on send SMS after entering the details
- Provide verification code received in your mobile
- Now select basic plan and click on complete sign up
- Now a screen appear as follows then click on AWS management console

- You will navigating to the aws page ..where you need to Select role as Academic/Researcher and interest as DevOps and click on submit
- Sign into AWS console as Root user
- A console screen appears aws will appear




**SBQ's**

1. Your Pod keeps restarting repeatedly. What will you do?

ANS: Check logs: kubectl logs <pod-name>

Describe pod: kubectl describe pod <pod-name>

Look for crash loop or configuration issues.

2. A Kubernetes pod is stuck in a "Pending" state. What could be the possible reasons, and how would you troubleshoot it?

ANS :
- Causes: No resources, nodeSelector mismatch, or PVC issues.
- Troubleshoot: `kubectl describe pod <pod-name>`

3. How would you debug a failed deployment in Kubernetes?

ANS : Check rollout status: kubectl rollout status deployment <deployment-name>

Check events: kubectl describe deployment <deployment-name>

4. You have a Kubernetes Deployment with multiple replicas, and some pods are failing health checks. How would you identify the root cause and fix it?

ANS : Run: kubectl describe pod <pod-name>

Inspect readiness/liveness probe errors.

Check container logs.

5. How do you roll back a faulty deployment?

ANS : kubectl rollout undo deployment <deployment-name>

6. How do you debug a running Pod?

ANS : kubectl exec -it <pod-name> -- /bin/bash

Or check logs: kubectl logs <pod-name>

7. You need to expose a local service externally.How to do?

ANS : kubectl expose deployment myapp --type=NodePort --port=80

minikube service myapp --url

8. How to Start and stop Nagios

ANS : docker start nagiosdemo

docker stop nagiosdemo

9. You installed Nagios but the web interface shows *"Unable to connect to Nagios process"*.how to resolve this?

ANS : Restart container.

Check configuration errors inside container logs.

10. You added a new host in Nagios, but it's not appearing on the web interface.how to check?

ANS : Check nagios.cfg and included configs.

Ensure config file syntax is correct.

Restart Nagios container.

11. How can you check whether Nagios is running properly?

ANS : docker ps

12. How do you view Nagios logs in real-time

ANS : docker logs -f nagiosdemo

13. What are the advantages of using Nagios?

ANS : Real-time monitoring

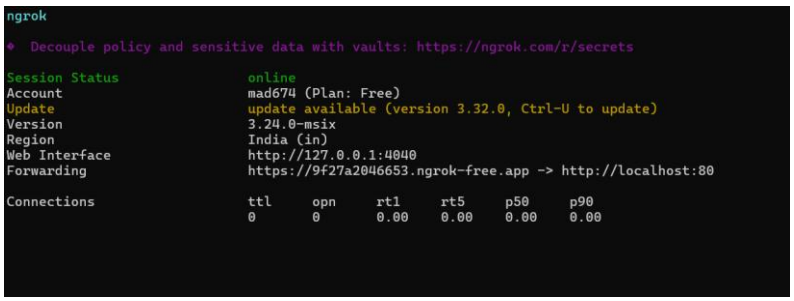
Custom alerting system

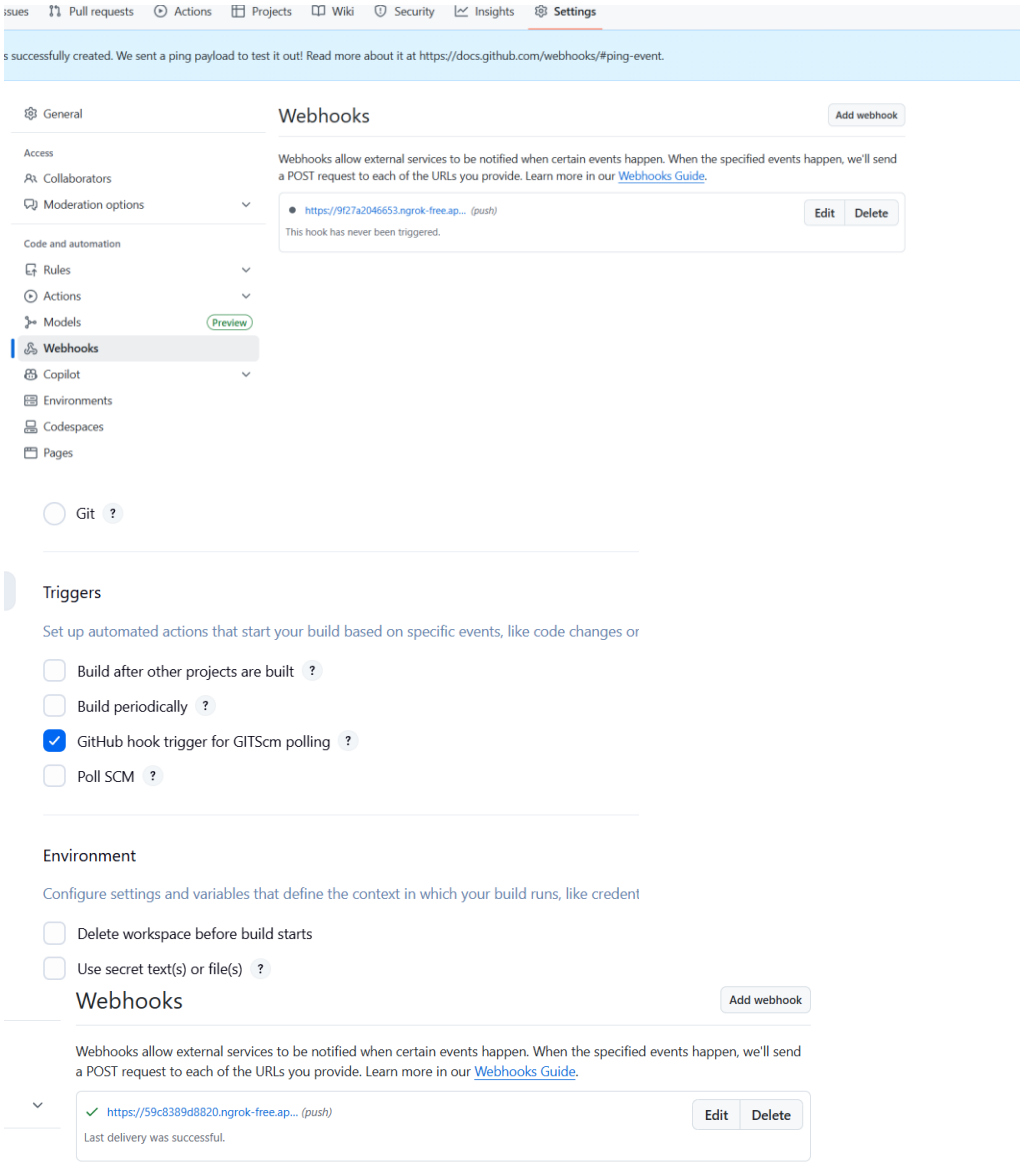
Supports plugins for extensibility

Easy visualization through Web UI

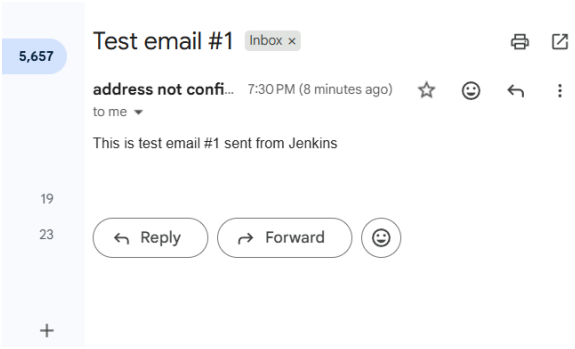
**Conclusion:** In this week we learnt how to deploy pods in minikube and how to monitor our local system using Nagios tool and will know how to create the aws free trier account.

# LAB REPORT – WEEK 11: Jenkins CI/CD Integration using GitHub Webhooks and Email Notifications



```
tarted by GitHub push by mad674
unning as SYSTEM
uilding in workspace C:\ProgramData\Jenkins\.jenkins\workspace\webhook
he recommended git tool is: NONE
o credentials specified
> git.exe rev-parse --resolve-git-dir
:\ProgramData\Jenkins\.jenkins\workspace\webhook\.git # timeout=10
etching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/mad674/ngrok.git # timeout=10
etching upstream changes from https://github.com/mad674/ngrok.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/mad674/ngrok.git
refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
hecking out Revision 63c2e22b0d237ad5b69eb259aba959016e148bcb
refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 63c2e22b0d237ad5b69eb259aba959016e148bcb # timeout=10
ommit message: "fi"
> git.exe rev-list --no-walk 2145a7540661aa3a018e1eaf050240059072eb31 # timeout=10
inished: SUCCESS
```




## Viva Questions & Answers

1. Q: What is Continuous Integration (CI)?

A: CI is the practice of automatically integrating code changes from multiple developers into a shared repository several times a day.

2. Q: What is Continuous Deployment (CD)?

A: CD automates the release of validated code to production after successful testing and integration.

3. Q: What is the role of Jenkins in a CI/CD pipeline?

A: Jenkins automates building, testing, and deploying code, ensuring a continuous integration and delivery workflow.

4. Q: What is a webhook in GitHub?

A: A webhook is an HTTP callback that notifies Jenkins when an event (like a push) occurs in a GitHub repository.

5. Q: Why are webhooks used in Jenkins integration?

A: They remove the need for periodic polling and allow immediate triggering of builds upon a GitHub event.

6. Q: What are the different types of build triggers available in Jenkins?

A: Build triggers include: Poll SCM, Build periodically, Trigger builds remotely, and GitHub webhook triggers.

7. Q: What is the difference between polling and webhook triggers?

A: Polling checks for changes at fixed intervals, while webhooks notify Jenkins instantly when changes occur.

8. Q: What is ngrok and why is it used in Jenkins–GitHub integration?

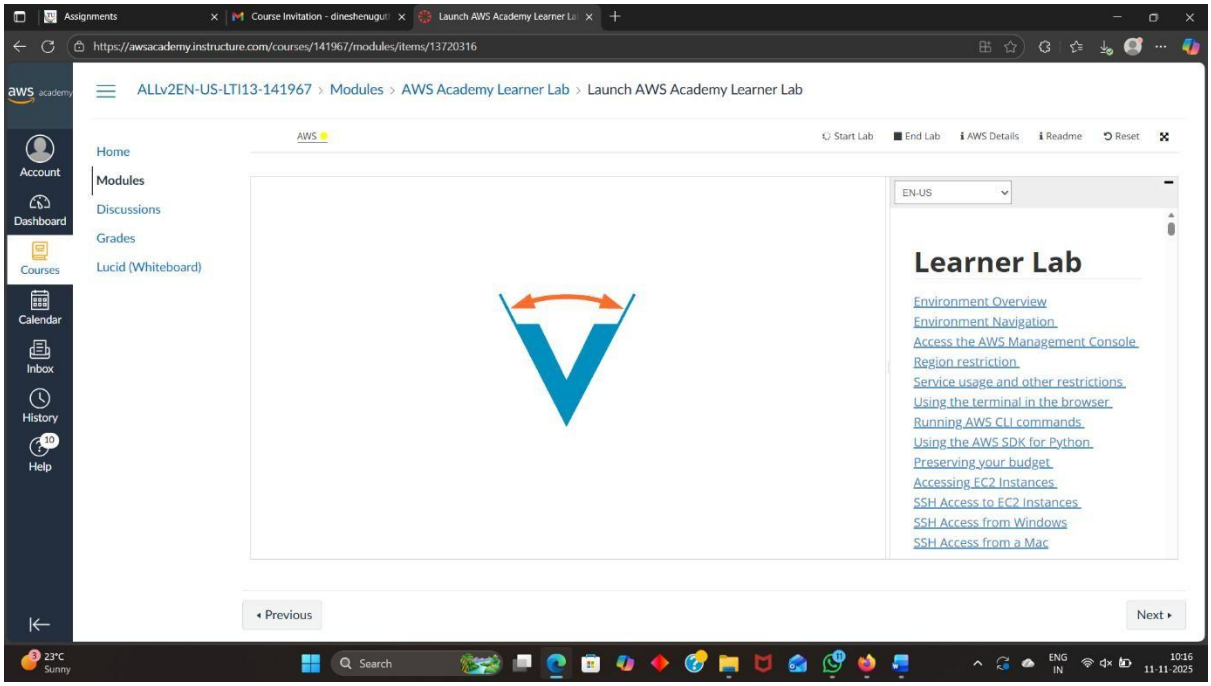
A: ngrok is a tunneling tool that exposes local servers to the internet, allowing GitHub to send webhooks to a local Jenkins instance.

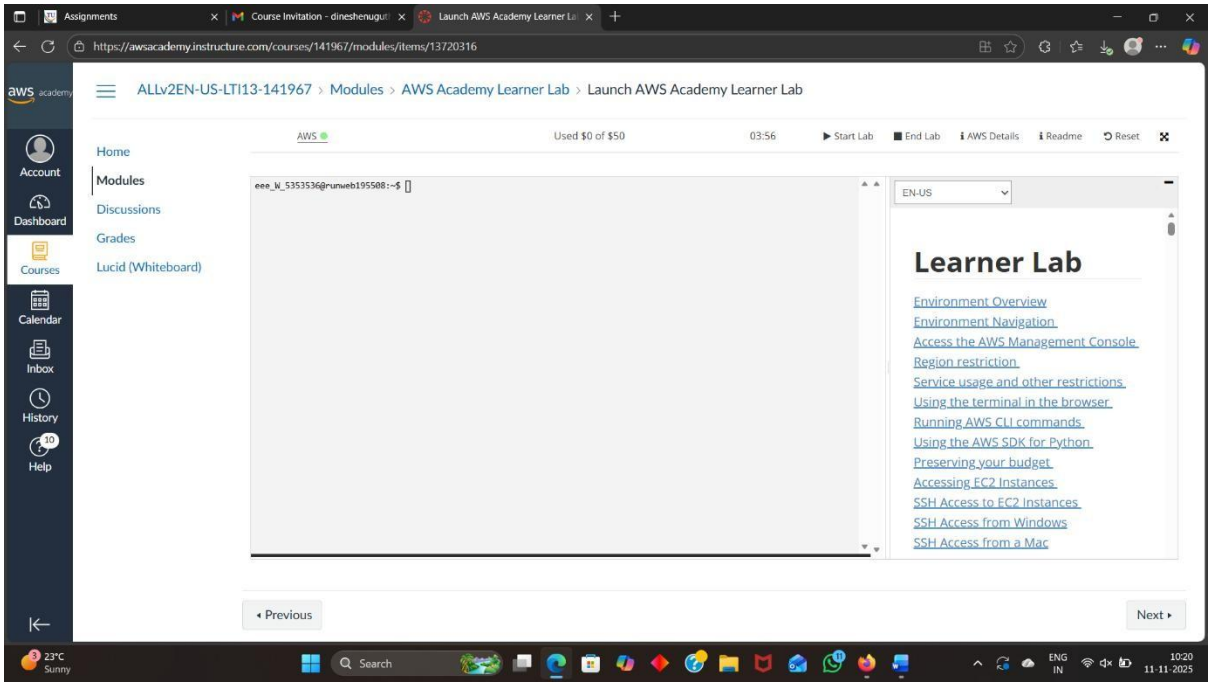
9. Q: How does ngrok help in setting up webhooks for Jenkins running locally?

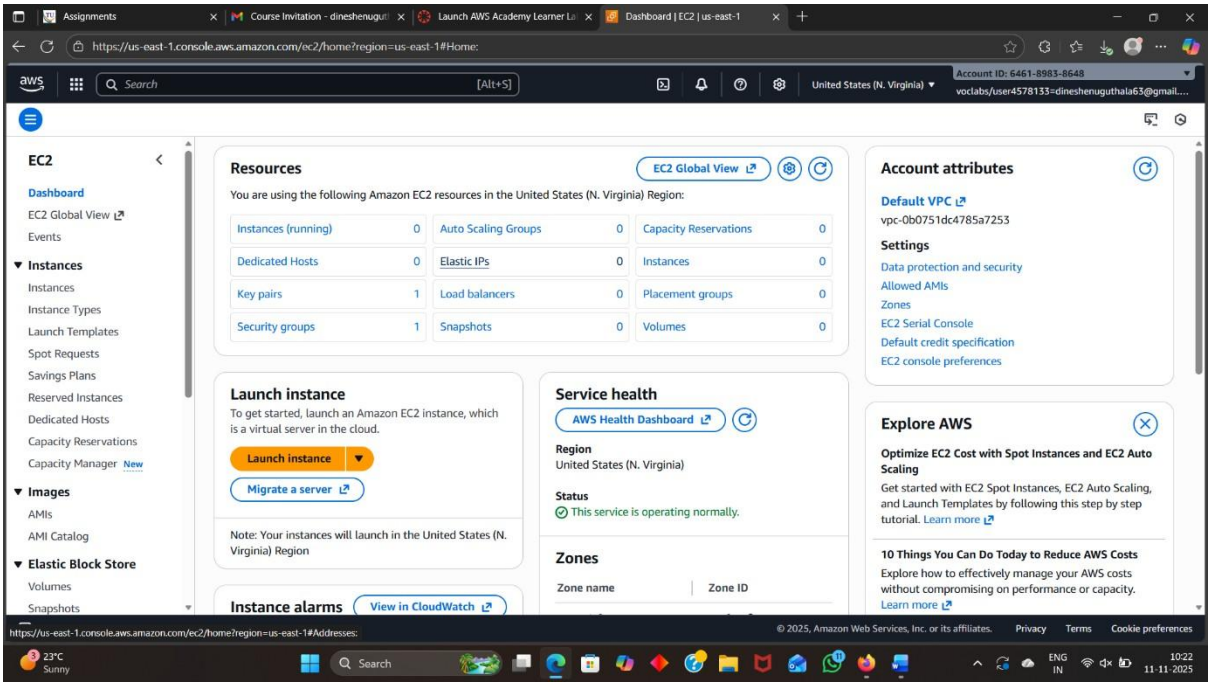
A: ngrok creates a public HTTPS URL that forwards requests to the local Jenkins, enabling GitHub to reach it.

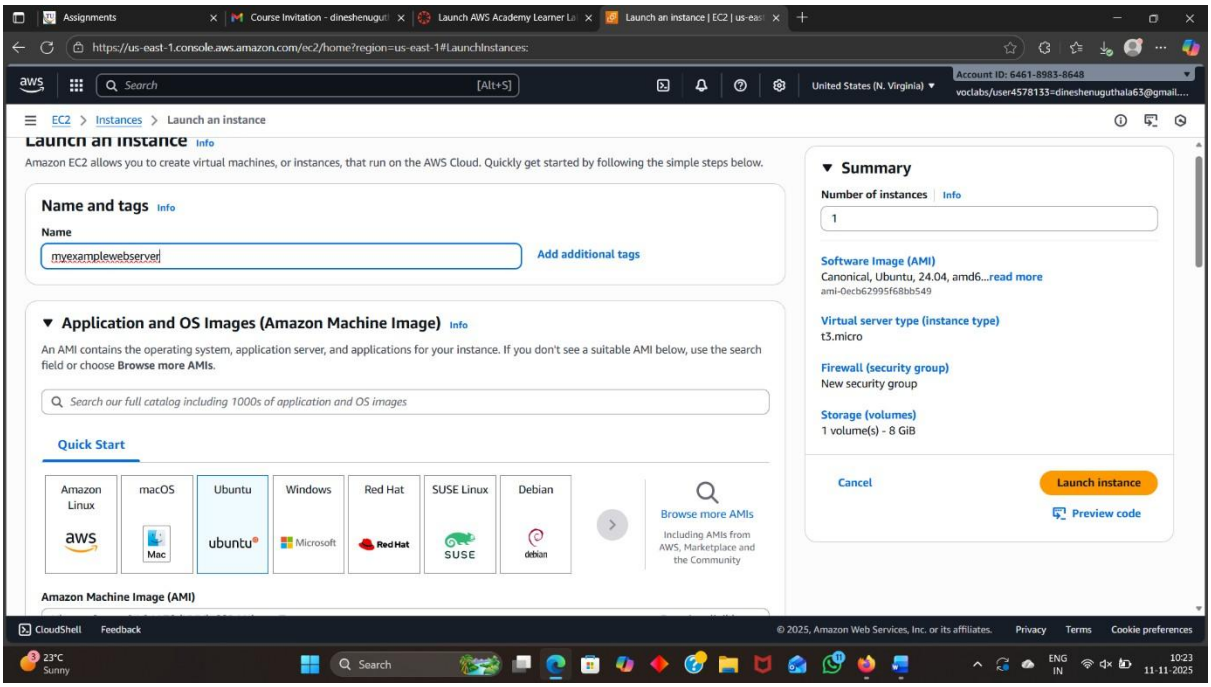
10. Q: Why do we configure email notifications in Jenkins?

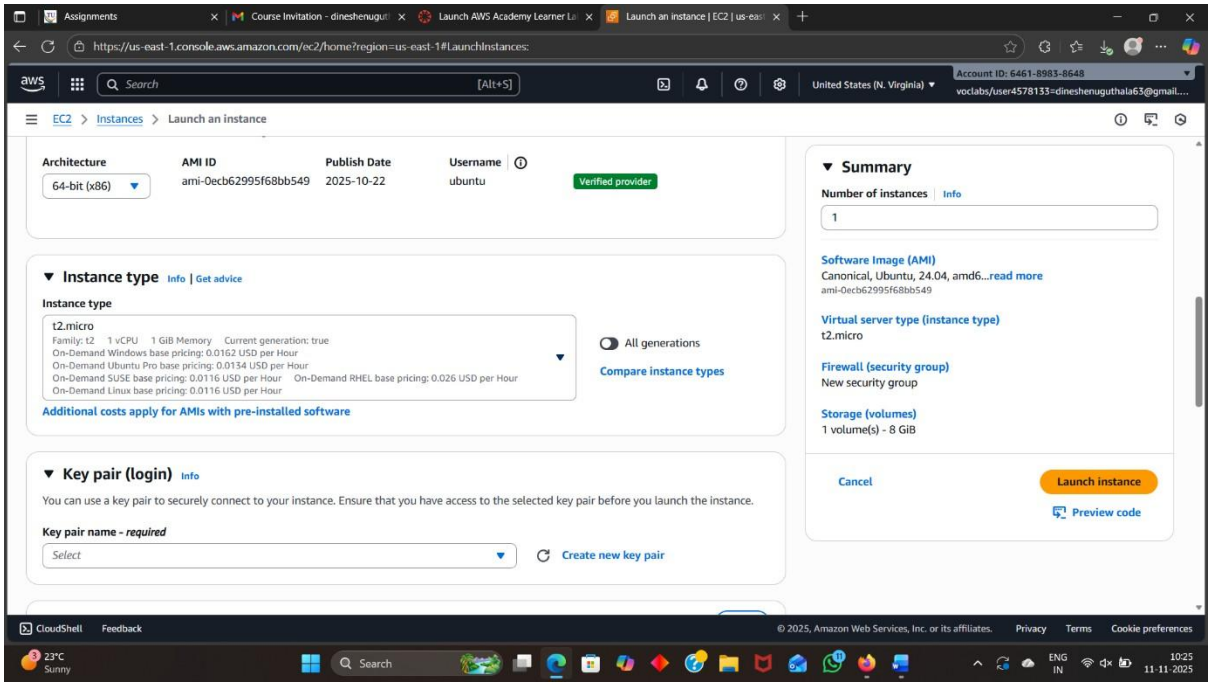
A: To receive automated build success or failure updates, helping developers monitor build status efficiently.

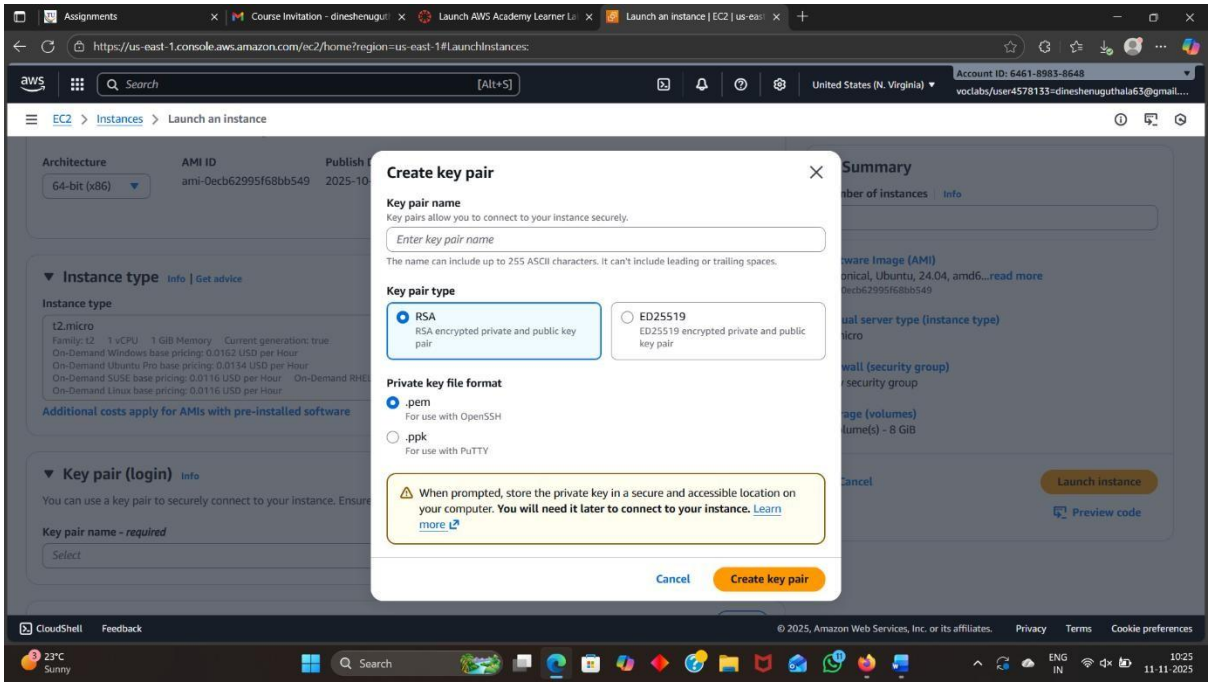

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
https://awsacademy.instructure.com/courses/141967/modules/items/13720316
ALLv2EN-US-LTI13-141967 › Modules › AWS Academy Learner Lab › Launch AWS Academy Learner Lab
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
AWS
Start Lab
End Lab
AWS Details
Readme
Reset
EN-US
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instances
SSH Access to EC2 Instances
SSH Access from Windows
SSH Access from a Mac
Previous
Next
23°C
Sunny
Search
ENG
IN
10:16
11-11-2025

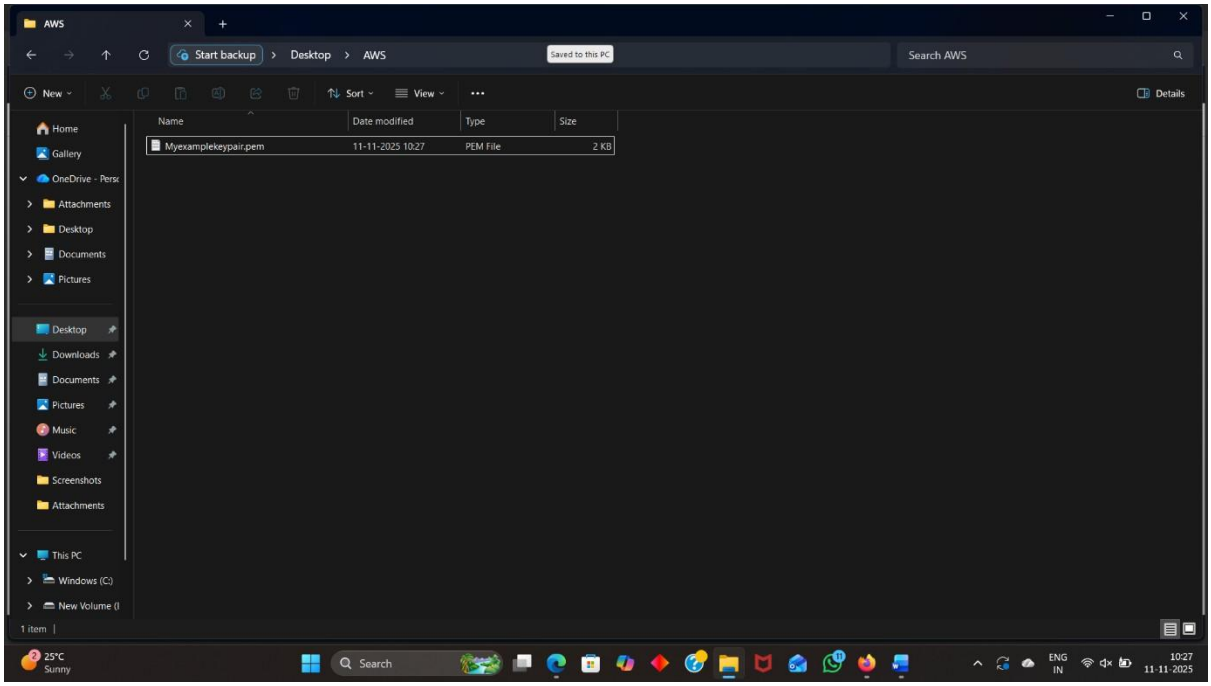

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
https://awsacademy.instructure.com/courses/141967/modules/items/13720316
ALLv2EN-US-LTI13-141967 › Modules › AWS Academy Learner Lab › Launch AWS Academy Learner Lab
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
AWS
Used $0 of $50
03:56
Start Lab
End Lab
AWS Details
Readme
Reset
EN-US
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instances
SSH Access to EC2 Instances
SSH Access from Windows
SSH Access from a Mac
Previous
Next
23°C
Sunny
Search
ENG
IN
10:20
11-11-2025

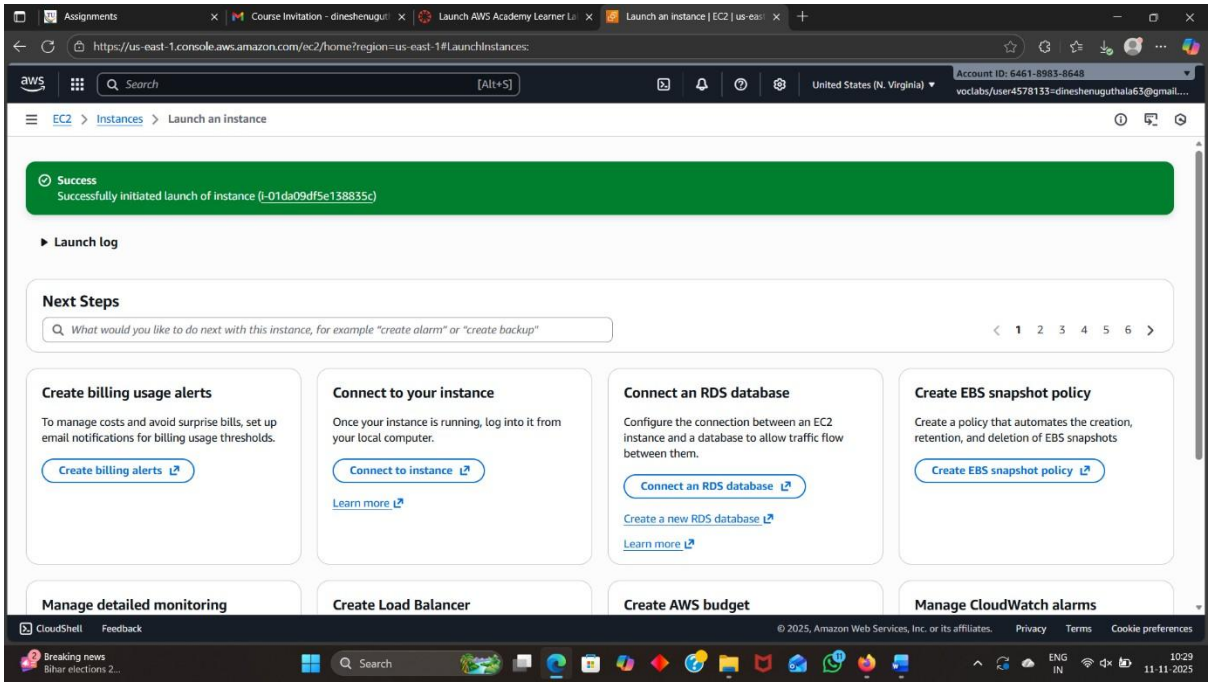

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
Dashboard | EC2 | us-east-1
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Home:
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Resources
EC2 Global View
You are using the following Amazon EC2 resources in the United States (N. Virginia) Region:
Instances (running) 0
Auto Scaling Groups 0
Capacity Reservations 0
Dedicated Hosts 0
Elastic IPs 0
Instances 0
Key pairs 1
Load balancers 0
Placement groups 0
Security groups 1
Snapshots 0
Volumes 0
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Migrate a server
Note: Your instances will launch in the United States (N. Virginia) Region
Instance alarms
View in CloudWatch
Service health
AWS Health Dashboard
Region
United States (N. Virginia)
Status
This service is operating normally.
Zones
Zone name
Zone ID
Account attributes
Default VPC
vpc-0b0751dc4785a7253
Settings
Data protection and security
Allowed AMIs
Zones
EC2 Serial Console
Default credit specification
EC2 console preferences
Explore AWS
Optimize EC2 Cost with Spot Instances and EC2 Auto Scaling
Get started with EC2 Spot Instances, EC2 Auto Scaling, and Launch Templates by following this step by step tutorial. Learn more
10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity.
Learn more
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Addresses:
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
23°C
Sunny
Search
ENG
IN
10:22
11-11-2025

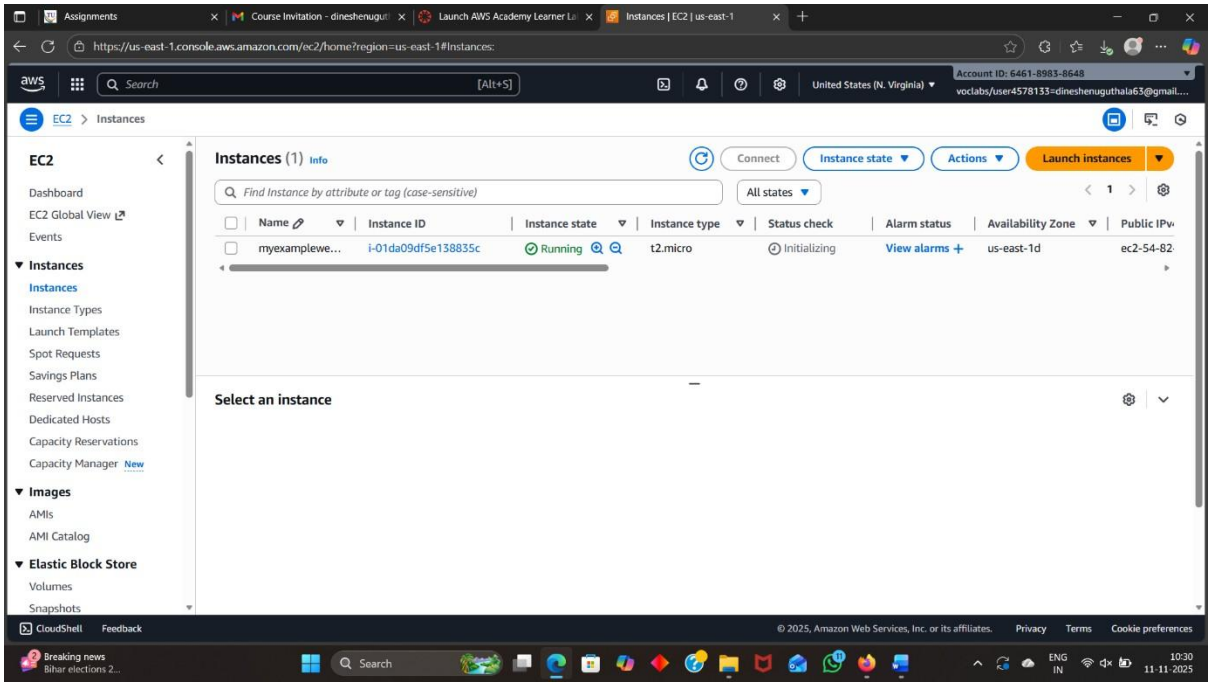

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
Launch an instance | EC2 | us-east
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
myexamplewebserver
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0ecb62995f68bb549
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
23°C
Sunny
Search
ENG
IN
10:23
11-11-2025

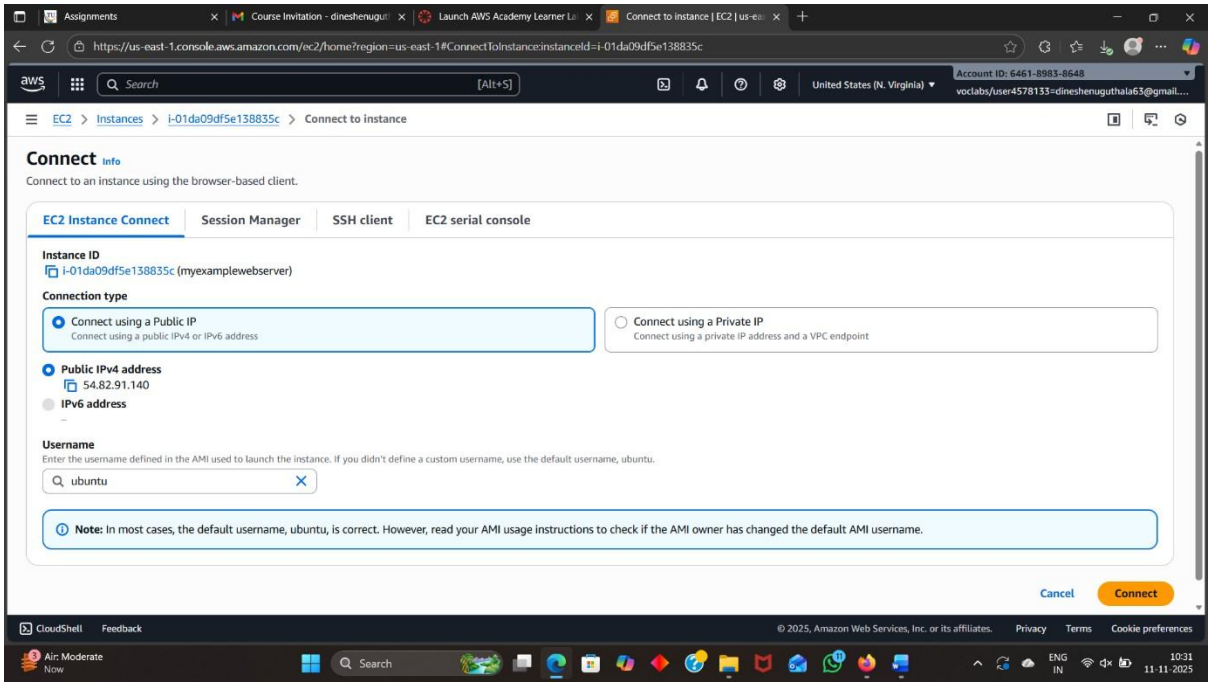

Assignments
Course Invitation - dineshenuguth
Launch AWS Academy Learner La
Launch an instance | EC2 | us-east
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search [Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances > Launch an instance
Architecture
64-bit (x86)
AMI ID
ami-0ecb62995f68bb549
Publish Date
2025-10-22
Username
ubuntu
Verified provider
Summary
Number of instances | Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0ecb62995f68bb549
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Instance type Info | Get advice
Instance type
t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Select
Create new key pair
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
23°C
Sunny
Search
ENG
IN
10:25
11-11-2025

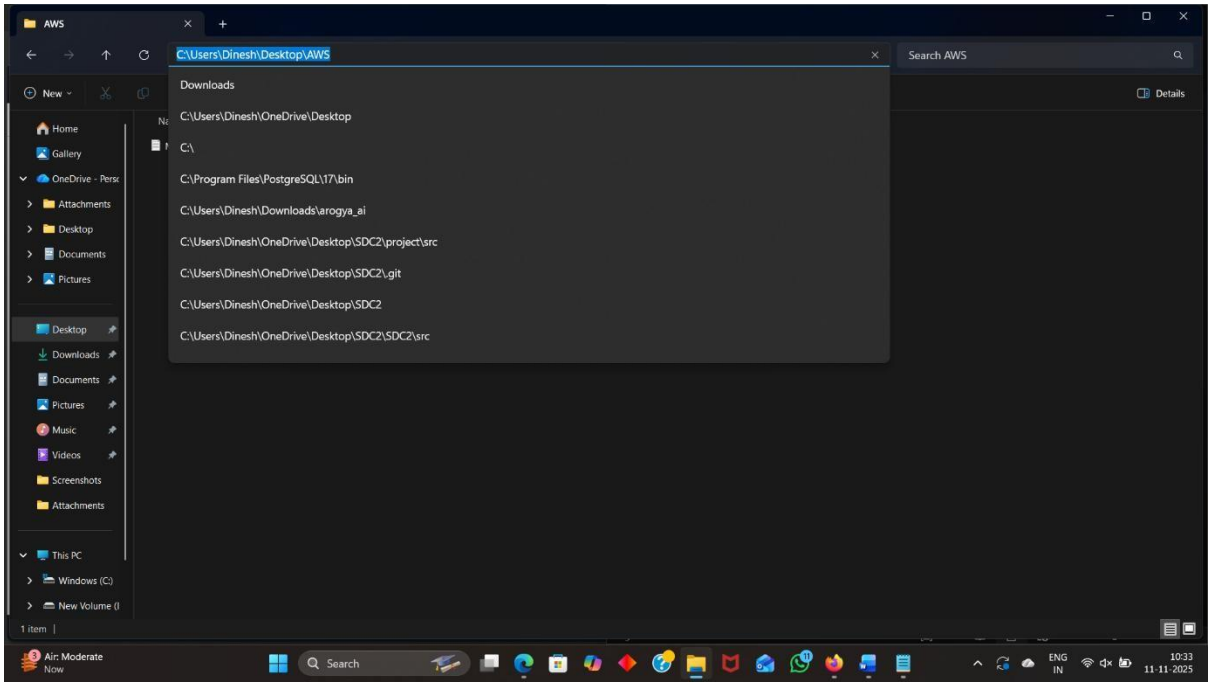

Assignments
Course Invitation - dineshenuguth
Launch AWS Academy Learner La
Launch an instance | EC2 | us-east
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search [Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances > Launch an instance
Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
Enter key pair name
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
Cancel
Create key pair
Summary
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
23°C
Sunny
Search
ENG
IN
10:25
11-11-2025

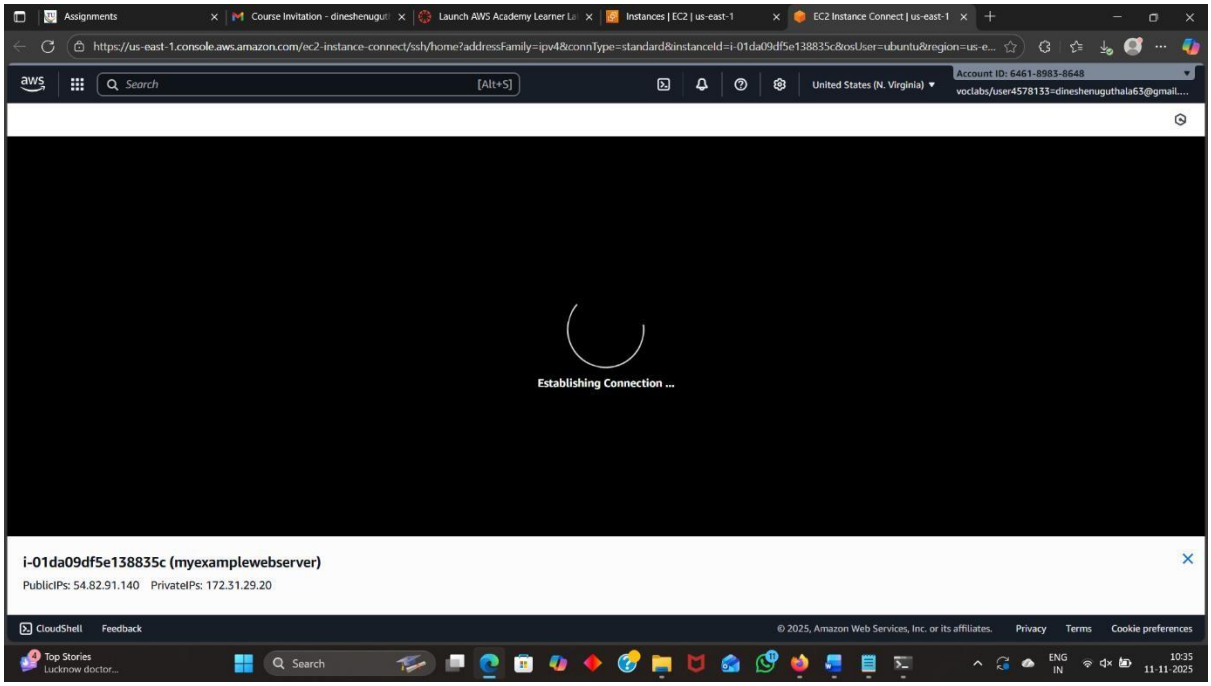

AWS
Start backup
Desktop
AWS
Saved to this PC
Search AWS
New
Sort
View
Details
Name
Date modified
Type
Size
Myexamplekeypair.pem
11-11-2025 10:27
PEM File
2 KB
1 item
25°C
Sunny
Search
ENG
IN
10:27
11-11-2025

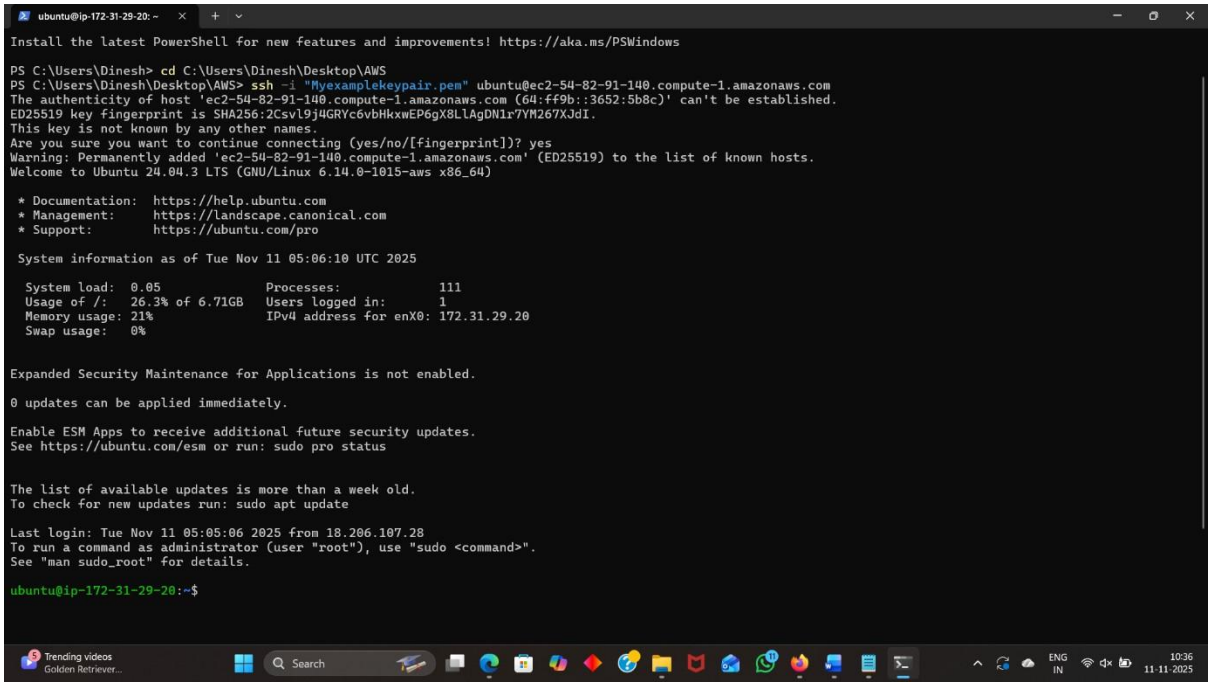

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
Launch an instance | EC2 | us-east
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-01da09df5e138835c)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
1 2 3 4 5 6
Create billing usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds.
Create billing alerts
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy
Manage detailed monitoring
Create Load Balancer
Create AWS budget
Manage CloudWatch alarms
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Breaking news
Bihar elections 2...
Search
ENG
IN
10:29
11-11-2025

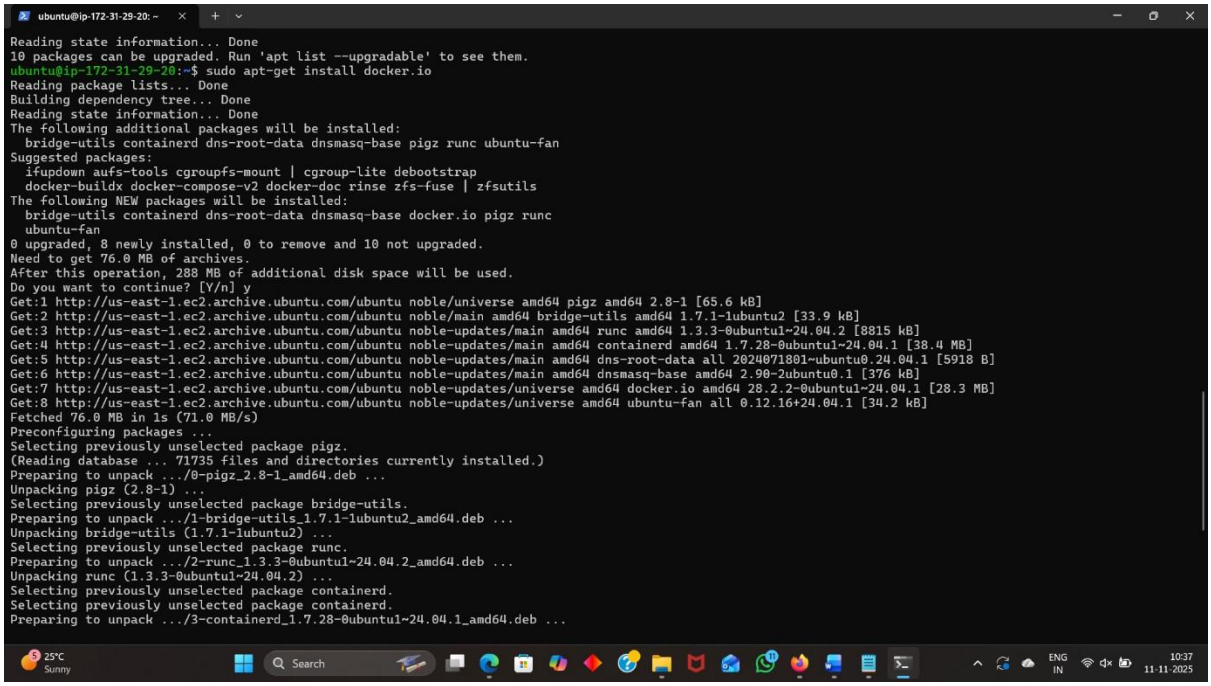

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
Instances | EC2 | us-east-1
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv
myexamplewe...
i-01da09df5e138835c
Running
t2.micro
Initializing
View alarms
us-east-1d
ec2-54-82
Select an instance
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Breaking news
Bihar elections 2...
Search
ENG
IN
10:30
11-11-2025

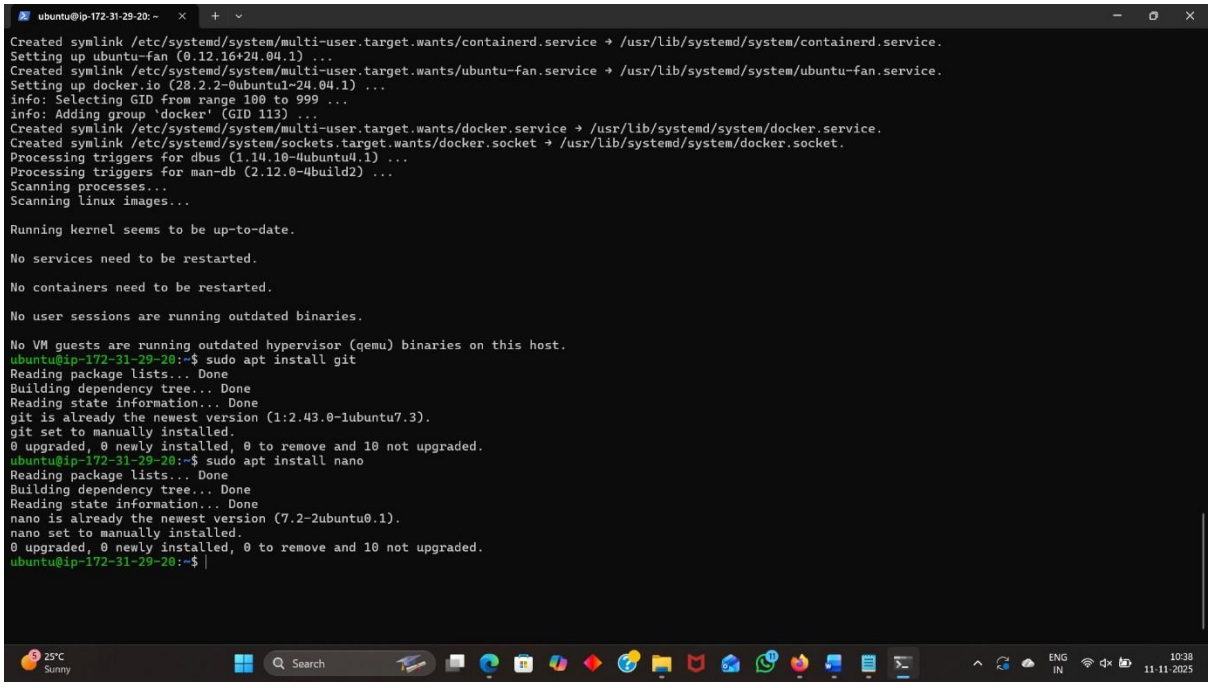

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
Connect to instance | EC2 | us-ea
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-01da09df5e138835c
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances > i-01da09df5e138835c > Connect to instance
Connect Info
Connect to an instance using the browser-based client.
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-01da09df5e138835c (myexamplewebserver)
Connection type
Connect using a Public IP
Connect using a public IPv4 or IPv6 address
Connect using a Private IP
Connect using a private IP address and a VPC endpoint
Public IPv4 address
54.82.91.140
IPv6 address
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu
Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Air: Moderate
Now
Search
ENG
IN
10:31
11-11-2025

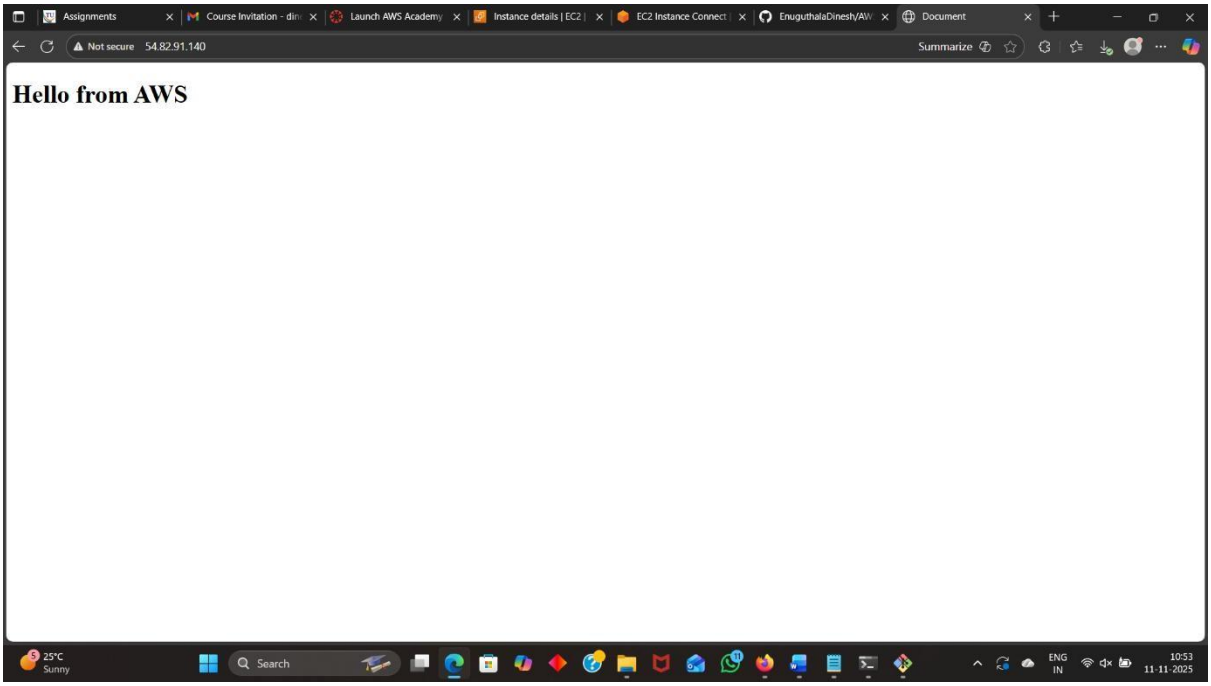

AWS
C:\Users\Dinesh\Desktop\AWS
Search AWS
Downloads
C:\Users\Dinesh\OneDrive\Desktop
C:\
C:\Program Files\PostgreSQL\17\bin
C:\Users\Dinesh\Downloads\arogya_ai
C:\Users\Dinesh\OneDrive\Desktop\SDC2\project\src
C:\Users\Dinesh\OneDrive\Desktop\SDC2\.git
C:\Users\Dinesh\OneDrive\Desktop\SDC2
C:\Users\Dinesh\OneDrive\Desktop\SDC2\SDC2\src
New
Details
Home
Gallery
Attachments
Desktop
Documents
Pictures
Downloads
Music
Videos
Screenshots
This PC
Windows (C:)
1 item
Search
ENG
IN
10:33
11-11-2025

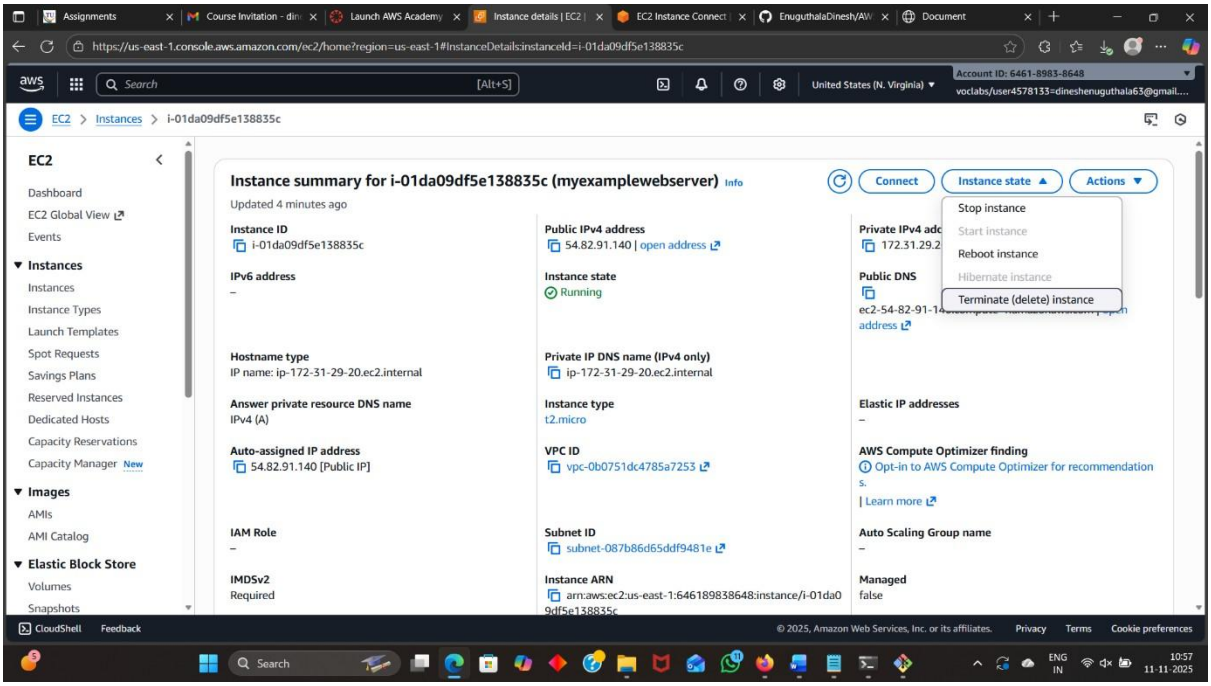

Assignments
Course Invitation - dineshenugut
Launch AWS Academy Learner La
Instances | EC2 | us-east-1
EC2 Instance Connect | us-east-1
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-01da09df5e138835c&osUser=ubuntu&region=us-e...
aws
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
Establishing Connection ...
i-01da09df5e138835c (myexamplewebserver)
PublicIPs: 54.82.91.140 PrivateIPs: 172.31.29.20
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Top Stories
Lucknow doctor...
Search
ENG
IN
10:35
11-11-2025

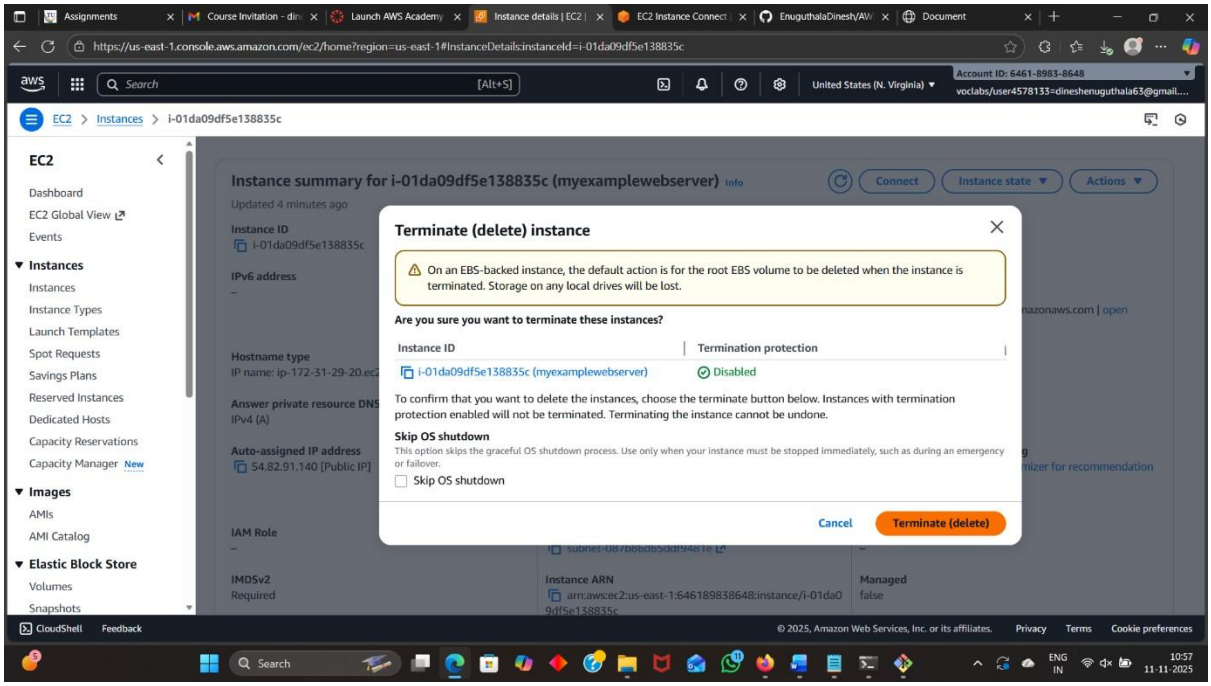

ubuntu@ip-172-31-29-20: ~
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Dinesh> cd C:\Users\Dinesh\Desktop\AWS
PS C:\Users\Dinesh\Desktop\AWS> ssh -i "Myexamplekeypair.pem" ubuntu@ec2-54-82-91-140.compute-1.amazonaws.com
The authenticity of host 'ec2-54-82-91-140.compute-1.amazonaws.com (64:ff9b::3652:5b8c)' can't be established.
ED25519 key fingerprint is SHA256:2Csvl9j4GRYc6vbHkxwEP6gX8LlAgDN1r7YM267XJdI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-82-91-140.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Tue Nov 11 05:06:10 UTC 2025
System load: 0.05 Processes: 111
Usage of /: 26.3% of 6.71GB Users logged in: 1
Memory usage: 21% IPv4 address for enX0: 172.31.29.20
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
Last login: Tue Nov 11 05:05:06 2025 from 18.206.107.28
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-29-20:~$
Trending videos
Golden Retriever...
Search
ENG
IN
10:36
11-11-2025

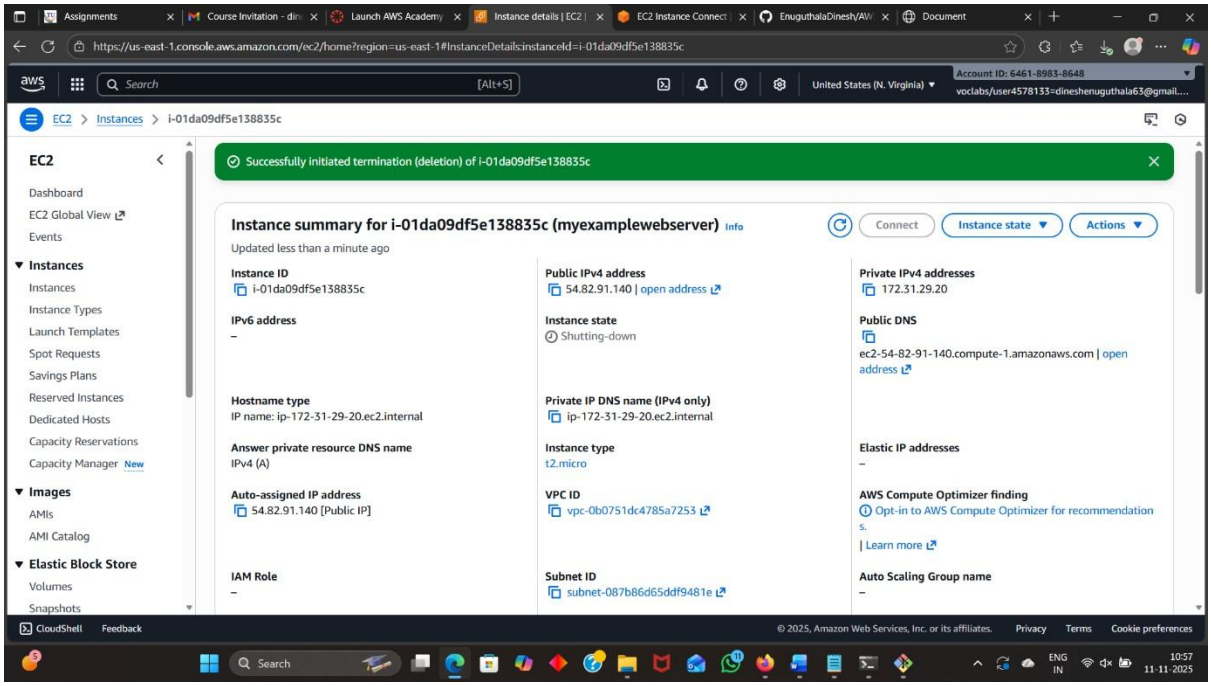

ubuntu@ip-172-31-29-20: ~
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-29-20:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional 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 10 not upgraded.
Need to get 76.0 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.2 [8815 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.1 [38.4 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801~ubuntu0.24.04.1 [5918 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.0 MB in 1s (71.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 71735 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.3.3-0ubuntu1~24.04.2_amd64.deb ...
Unpacking runc (1.3.3-0ubuntu1~24.04.2) ...
Selecting previously unselected package containerd.
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.28-0ubuntu1~24.04.1_amd64.deb ...
25°C
Sunny
Search
ENG
IN
10:37
11-11-2025

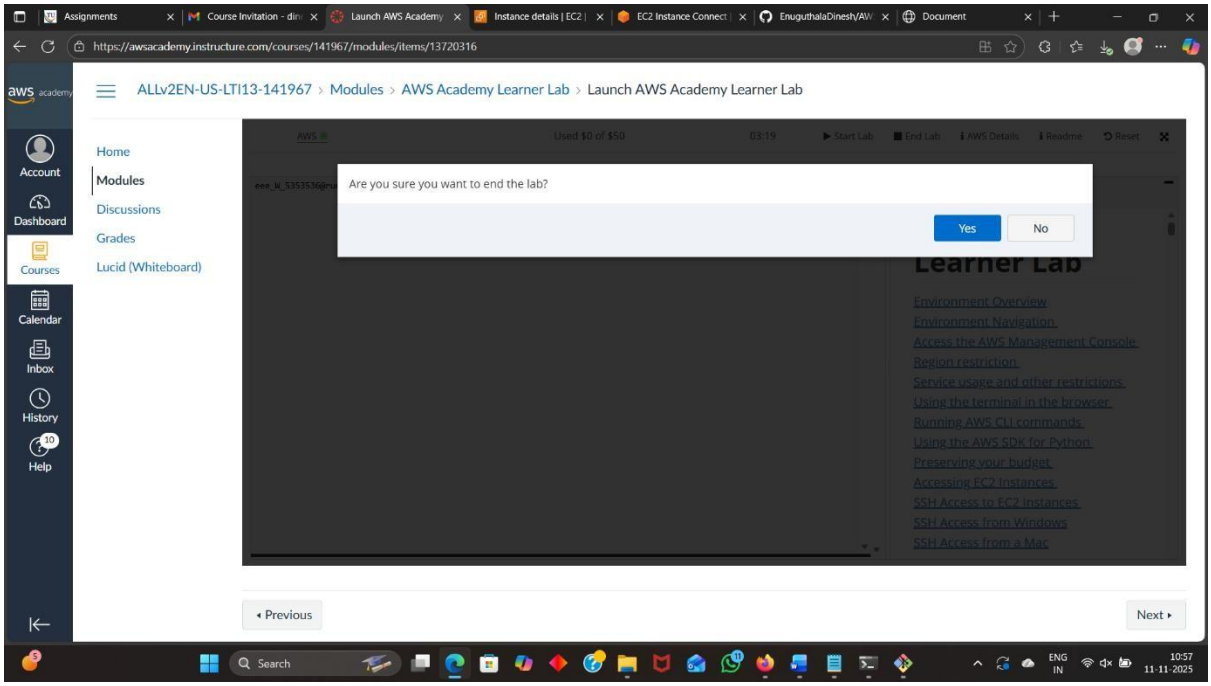

ubuntu@ip-172-31-29-20: ~
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16+24.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (28.2.2-0ubuntu1~24.04.1) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group 'docker' (GID 113) ...
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 (1.14.10-4ubuntu4.1) ...
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-172-31-29-20:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
ubuntu@ip-172-31-29-20:~$ 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 10 not upgraded.
ubuntu@ip-172-31-29-20:~$
25°C
Sunny
Search
ENG
IN
10:38
11-11-2025


Assignments
Course Invitation - din
Launch AWS Academy
Instance details | EC2 |
EC2 Instance Connect
EnuguthalaDinesh/AW
Document
Not secure 54.82.91.140
Summarize
Hello from AWS
25°C
Sunny
Search
ENG
IN
10:53
11-11-2025


Assignments
Course Invitation - din
Launch AWS Academy
Instance details | EC2 |
EC2 Instance Connect
EnuguthalaDinesh/AW
Document
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-01da09df5e138835c
Search
[Alt+S]
United States (N. Virginia)
Account ID: 6461-8983-8648
voclabs/user4578133=dineshenuguthala63@gmail....
EC2 > Instances > i-01da09df5e138835c
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Instance summary for i-01da09df5e138835c (myexamplewebserver) Info
Updated 4 minutes ago
Connect
Instance state
Actions
Stop instance
Start instance
Reboot instance
Hibernate instance
Terminate (delete) instance
Instance ID
i-01da09df5e138835c
Public IPv4 address
54.82.91.140 | open address
Private IPv4 add
172.31.29.2
IPv6 address
–
Instance state
Running
Public DNS
address
Hostname type
IP name: ip-172-31-29-20.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-29-20.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
–
Auto-assigned IP address
54.82.91.140 [Public IP]
VPC ID
vpc-0b0751dc4785a7253
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
Learn more
IAM Role
–
Subnet ID
subnet-087b86d65ddf9481e
Auto Scaling Group name
–
IMDSv2
Required
Instance ARN
arn:aws:ec2:us-east-1:646189838648:instance/i-01da09df5e138835c
Managed
false
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
10:57
11-11-2025


Terminate (delete) instance
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
Termination protection
i-01da09df5e138835c (myexamplewebserver)
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.
Skip OS shutdown
This option skips the graceful OS shutdown process. Use only when your instance must be stopped immediately, such as during an emergency or failover.
Skip OS shutdown
Cancel
Terminate (delete)


Successfully initiated termination (deletion) of i-01da09df5e138835c
Instance summary for i-01da09df5e138835c (myexamplewebserver)
Updated less than a minute ago
Instance ID
i-01da09df5e138835c
Public IPv4 address
54.82.91.140 | open address
Private IPv4 addresses
172.31.29.20
IPv6 address
–
Instance state
Shutting-down
Public DNS
ec2-54-82-91-140.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-29-20.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-29-20.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
–
Auto-assigned IP address
54.82.91.140 [Public IP]
VPC ID
vpc-0b0751dc4785a7253
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
–
Subnet ID
subnet-087b86d65ddf9481e
Auto Scaling Group name
–




Scenario-Based Questions and Answers

**1. You have a simple index.html file on your laptop and you launched an EC2 instance with Amazon Linux 2. The instance is running but when you open the public IP in browser, the page doesn't load. What steps will you take to host the index.html?**
**Answer:** Install Apache → Start service → Copy index.html to /var/www/html/ → Allow port 80 in Security Group.

**2. You deployed your index.html to /var/www/html/ directory on EC2, but the web page still isn't loading. What are two possible issues you would check?**
**Answer:** Check if Apache is running and if port 80 is open in Security Group.

**3. You installed Apache HTTP server on EC2 to host index.html, but the service stops after instance reboot. What command should you run to ensure it auto-starts on boot?**
**Answer:** sudo systemctl enable httpd

**4. You are deploying a Maven web application onto an EC2 instance. Maven is not installed on the instance. What commands or steps will you follow to install Maven on Amazon Linux/Ubuntu EC2?**
**Answer:**

- Amazon Linux: sudo yum install maven -y
- Ubuntu: sudo apt update && sudo apt install maven -y

**5. You built a Maven project on EC2, and a .war file generated inside target/. You want to deploy it using Tomcat. Where will you place the .war file and why?**
**Answer:** Place it in /usr/share/tomcat/webapps/ — Tomcat auto-deploys .war files from there.

**6. Your Maven web app is deployed to Tomcat on EC2, but accessing it via browser gives 404 error. What configuration or path issues will you check?**
**Answer:** Check if .war deployed correctly, Tomcat logs, context path, and service status.

**7. You can access your web application locally on the EC2 instance using curl localhost:8080 but not from your browser. What AWS setting is likely missing or misconfigured?**
**Answer:** Security Group missing inbound rule for port 8080.

**8. You have deployed your web app on EC2 successfully, but after shutting down your local Wi-Fi and reconnecting, the public IP changed and the app is not opening. What AWS feature helps avoid this issue?**
**Answer:** Use an Elastic IP.

**9. You want to automate deployment of your index.html whenever you restart the EC2 instance. Which EC2 feature can be used to run commands automatically during instance setup?**
**Answer:** Use User Data script.

**10. Your Maven application needs external dependencies during build but EC2 has no internet. What AWS service or change can allow the EC2 instance to download dependencies securely?**
**Answer:** Use a NAT Gateway or VPC Endpoint for internet access.