

Roll Number: C24079

MCAL23 DevOps Lab INDEX				
Sr.No	Title	CO	Date	Sign
1	Demonstrate basic Git commands	CO2		
2	Create and fork repositories in Git Hub. Apply branch, merge and rebase concepts.	CO2		
3	Demonstrate Git for Collaboration	CO2		
4	Demonstrate Collaborating and cloning using Git	CO2		
5	Using GitLab Web IDE	CO2		
6	Demonstrate CI/CD Workflow in GitLab using .py, .bash, .java file	CO2		
7	Demonstrate setting Jenkins CI/CD pipeline.	CO3		
8	Demonstrate Setting up of a CI/CD pipeline to build and deploy a web application to a local HTTP server	CO3		
9	Demonstrate basic Docker commands	CO3		
10	Develop a simple containerized application using Docker	CO3		
11	Demonstrate add-on ansible commands	CO4		
12	Demonstrate Ansible Playbooks	CO4		

PRACTICAL 1**AIM: Demonstrate basic git commands**

1. Set-up Git (One-time setup)

- **git config --global user.name "Anil"** -- Set your name for commits.
- **git config --global user.email "prajapatianil2003@gmail.com"** -- Set your email for commits.

```
anant@Anant MINGW64 /d/dev/prac
$ git config --global user.name "anant"

anant@Anant MINGW64 /d/dev/prac
$ git config --global user.email "anantkumbhar5835@gmail.com"
```

2. Start a new project

- **git init** :- Initialize a new repository in your project folder.

```
anant@Anant MINGW64 /d/dev/prac
$ git init
Initialized empty Git repository in D:/dev/prac/.git/
```

Name	Date modified	Type	Size
.git	20-05-2025 16:14	File folder	

- **git status**:- Check the status of files in the working directory (untracked, modified, etc.).

```
anant@Anant MINGW64 /d/dev/prac (master)
$ git status
On branch master

No commits yet

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

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

- **git add .** :- Stage all changes (new, modified, or deleted files).
- **git commit -m "Initial Commit"** :- Commit the staged files with a message describing the changes.

```
anant@Anant MINGW64 /d/dev/prac (master)
$ git add .

anant@Anant MINGW64 /d/dev/prac (master)
$ git commit -m "Initial Commit"
[master (root-commit) 80e3826] Initial Commit
 1 file changed, 1 insertion(+)
  create mode 100644 message.txt
```

3. Work on a New Feature

- **git branch <branch_name>** :- Create a new branch for a feature (e.g., feature-1).

- **git checkout <branch_name>** :- Switch to the new branch.

```
anant@Anant MINGW64 /d/dev/prac (master)
$ git branch new-feature

anant@Anant MINGW64 /d/dev/prac (master)
$ git checkout new-feature
Switched to branch 'new-feature'
```

4. Make Changes and Commit them

- **git add <file>**:- Stage specific files for commit.
- **git commit -m "Add new feature"** :- Commit changes with a descriptive message.
- **git log** :- View the history of commits in the current branch.

```
anant@Anant MINGW64 /d/dev/prac (new-feature)
$ git add message1.txt

anant@Anant MINGW64 /d/dev/prac (new-feature)
$ git status
On branch new-feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   message1.txt

anant@Anant MINGW64 /d/dev/prac (new-feature)
$ git commit -m "Added new feature"
[new-feature 8de5f0c] Added new feature
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 message1.txt

anant@Anant MINGW64 /d/dev/prac (new-feature)
$ git log
commit 8de5f0cc92eff764ae346869fbe42a006889c55c (HEAD -> new-feature)
Author: anant <anantkumbhar5835@gmail.com>
Date:   Tue May 20 16:06:29 2025 +0530

    Added new feature

commit 80e38260503540590a4823fe1b3dd5d1ad66da21 (master)
Author: anant <anantkumbhar5835@gmail.com>
Date:   Tue May 20 15:57:16 2025 +0530

    Initial Commit
```

5. Merge changes back into main branch.

- **git checkout main** :- Switch back to the main branch
- **git merge <branch_name>** :- Merge the changes from the feature branch into the main branch.

```

anant@Anant MINGW64 /d/dev/prac (new-feature)
$ git checkout master
Switched to branch 'master'

anant@Anant MINGW64 /d/dev/prac (master)
$ git merge new-feature
Updating 80e3826..8de5f0c
Fast-forward
 message1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 message1.txt

```

6. Set up a remote repository

- **git remote add origin <repository-url>** :- Link your local repository to a remote repository (e.g., on GitHub).
- **git push -u origin main**
Push the main branch to the remote repository for the first time.

```

anant@Anant MINGW64 /d/dev/prac (master)
$ git remote add origin "https://github.com/AnantKumbhar/dev.git"

anant@Anant MINGW64 /d/dev/prac (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/AnantKumbhar/dev.git'

anant@Anant MINGW64 /d/dev/prac (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 463 bytes | 463.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AnantKumbhar/dev.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

7. Collaborate with a team

- **git pull origin main** :- Fetch and merge changes from the remote main branch to your local branch.
- **git branch**:- List all branches to see if new ones were created by collaborators.

```
anant@Anant MINGW64 /d/dev/prac (master)
$ git pull origin master
From https://github.com/AnantKumbhar/dev
 * branch            master      -> FETCH_HEAD
Already up to date.

anant@Anant MINGW64 /d/dev/prac (master)
$ git branch
* master
  new-feature
```

- **git fetch**:- Download updates from the remote without merging them.
- **git diff** :- Compare changes between your working directory and the staging area, or between commits.
- **git push** :- Push your committed changes to the remote branch

```
anant@Anant MINGW64 /d/dev/prac (master)
$ git fetch

anant@Anant MINGW64 /d/dev/prac (master)
$ git diff

anant@Anant MINGW64 /d/dev/prac (master)
$ git diff
diff --git a/message.txt b/message.txt
index 4258f5c..c7d45fa 100644
--- a/message.txt
+++ b/message.txt
@@ -1 +1 @@
-helloo
\ No newline at end of file
+helloo my name is anant kumbhar
\ No newline at end of file

anant@Anant MINGW64 /d/dev/prac (master)
$ git push
Everything up-to-date
```

PRACTICAL 2

AIM: Create and fork repositories in Git Hub. Apply branch, merge and rebase concepts.

Step 1: Initial Setup

1. **Create a Git repository (if not already created)**

If you don't have a repository yet, you can create one by running:

```
git init
anant@Anant MINGW64 /d/dev/prac (master)
$ git init
Reinitialized existing Git repository in D:/dev/prac/.git/
```

2. **Clone an existing repository (if you're working on an existing project)**

If you're working with an existing remote repository, you can clone it by running:

```
git clone <repository-url>
cd <repository-name>
anant@Anant MINGW64 /d/dev/prac (master)
$ git clone https://github.com/AnantKumbhar/dev.git
Cloning into 'dev'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.

anant@Anant MINGW64 /d/dev/prac (master)
$ cd dev
```

Step 2: Working with Branches

1. **Check the current branch** By default, Git starts with a branch named `main` or `master`. To see which branch you are currently on, use:

```
git branch
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git branch
* branch1
  master
  new_branch
```

The current branch will be marked with an asterisk (*).

2. Create a new branch To create a new branch, use:

```
git branch <branch-name>
anant@Anant MINGW64 /d/dev/prac/dev (master)
$ git branch new_branch
```

This creates the branch but does **not** switch to it.

3. Switch to the new branch To start working on the new branch, use:

```
git checkout <branch-name>
```

You can also combine the creation and switch into one command:

```
git checkout -b <branch-name>
anant@Anant MINGW64 /d/dev/prac/dev (master)
$ git checkout new_branch
Switched to branch 'new_branch'

anant@Anant MINGW64 /d/dev/prac/dev (new_branch)
$ git checkout -b branch1
Switched to a new branch 'branch1'
```

4. View all branches To see all branches in your repository:

```
git branch
```

The current branch will be marked with an asterisk (*).

```
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git branch
* branch1
  master
  new_branch
```

Step 3: Make Changes in the Branch

1. Make some changes in the code

Now that you're on your new branch, make some changes to your files (e.g., modify code, add new features, etc.).

2. Stage the changes After making changes, you need to add them to the staging area:

```
bash
```

```
git add <file-name> # Add a specific file
git add .           # Add all files (recommended if you want to stage everything)
```

```
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git add message.txt
```

```
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git add .
```

```
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git status
On branch branch1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:  new.txt
```

3. **Commit the changes** After staging, commit the changes to your branch:

```
git commit -m "Description of changes"
git
```

```
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git commit -m "Added some text"
[branch1 db66bfa] Added some text
 1 file changed, 1 insertion(+)
 create mode 100644 new.txt
```

Step 4: Merge the Branch into Main

1. **Switch to the `main` branch (or the branch you want to merge into)** Before merging, switch back to the `main` branch:

```
git checkout main
```

```
anant@Anant MINGW64 /d/dev/prac/dev (branch1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

2. **Pull the latest changes** Ensure your `main` branch is up to date with the remote repository:

```
git pull origin main
```

```
anant@Anant MINGW64 /d/dev/prac/dev (master)
$ git pull origin master
From https://github.com/AnantKumbhar/dev
 * branch            master      -> FETCH_HEAD
Already up to date.
```

- Merge the feature branch into `main` Now merge your branch into `main`:

`git merge <branch-name>`

- If there are no conflicts, Git will automatically complete the merge and add a merge commit.
- If there are conflicts, Git will notify you, and you'll need to resolve them manually.

```
anant@Anant MINGW64 /d/dev/prac/dev (master)
$ git merge branch1
Updating 8de5f0c..745150a
Fast-forward
```

Step 5: Resolving Merge Conflicts (If Any)

- Check for conflicts** If Git encounters conflicts during the merge, it will pause and mark the conflicted files.
 - Open the conflicted files** Conflicted sections will be marked with:
- ```
<<<<< HEAD
(changes from `main` branch)
=====
(changes from `<branch-name>`)
>>>>> <branch-name>
```
- Resolve the conflicts** Edit the file to keep the changes you want and remove the conflict markers (`<<<<<`, `=====`, `>>>>>`).
  - Mark the conflicts as resolved** After resolving conflicts, stage the files as resolved:

`git add <resolved-file>`

- Complete the merge** Once all conflicts are resolved, commit the merge:

`git commit`

Git will automatically create a merge commit if you didn't need to resolve conflicts manually.

## Step 6: Push Changes to Remote

- Push the changes to the remote repository** After merging, push the changes to the remote repository:

`git push origin main`

This updates the remote repository with the changes from your merge.

```
anant@Anant MINGW64 /d/dev/prac/dev (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 464 bytes | 464.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AnantKumbhar/dev.git
 8de5f0c..745150a master -> master
```

## Step 7: Clean Up (Optional)

- Delete the branch after merging (optional)** After merging, you can delete your feature branch if you no longer need it:

```
git branch -d <branch-name> # Deletes the local branch
anant@Anant MINGW64 /d/dev/prac/dev (master)
$ git branch -d branch1
Deleted branch branch1 (was 745150a).
```

- Delete the remote branch (optional)** If you want to delete the branch on the remote as well, use:

```
git push origin --delete <branch-name>
```

## Step 8: Regular Maintenance

- Sync your local repository with the remote regularly** To avoid conflicts, it's good practice to frequently pull changes from the main branch into your working branch:

```
git checkout <branch-name> # Switch to your feature branch
git pull origin main # Pull the latest changes from main
```

- Stay organized**

- Use descriptive branch names (e.g., feature/auth, bugfix/login).
- Regularly merge back into `main` to keep your changes synchronized.

## Practical 3 & 4

### AIM: Demonstrate Git for collaboration and cloning using Git

#### Set Up Git and GitHub

Before you start collaborating or cloning repositories, make sure you have the following set up:

1. **Install Git:** If you haven't already, download and install [Git](#) on your machine.
2. **Create a GitHub Account:** Go to [GitHub](#) and create an account if you don't have one.
3. **Configure Git:** Set up your Git configuration with your name and email.

```
git config --global user.name "Your Name"
```

```
git config --global user.email youremail@example.com
```

```
anant@Anant MINGW64 /d/dev/prac/dev (new_branch)
$ git config --global user.name "AnantKumbhar"

anant@Anant MINGW64 /d/dev/prac/dev (new_branch)
$ git config --global user.email anantkumbhar5835@gmail.com
```

#### Step 2: Clone a GitHub Repository

Cloning a repository allows you to create a copy of a project on your local machine, enabling you to work on it.

1. **Find a Repository to Clone:** Visit the repository page on GitHub (e.g., <https://github.com/username/repository>) and click the green **Code** button.
2. **Copy the Clone URL:** In the popup, choose either **HTTPS** or **SSH** and copy the URL. If you're using HTTPS, it will look like `https://github.com/username/repository.git`.
3. **Clone the Repository Locally:**

Open a terminal on your computer and navigate to the directory where you want to clone the repository. Then run:

```
git clone https://github.com/username/repository.git
```

- Replace `https://github.com/username/repository.git` with the URL you copied.

This will create a local copy of the repository on your machine.

- **Navigate to the Repository Folder:**

```
cd repository # Navigate into the cloned directory
```

```
anant@Anant MINGW64 /d/dev/prac/dev (new_branch)
$ git clone https://github.com/AnantKumbhar/dev.git
Cloning into 'dev'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 10 (delta 2), reused 8 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.

anant@Anant MINGW64 /d/dev/prac/dev (new_branch)
$ cd dev
```

### Step 3: Work on the Project Locally

Once you've cloned the repository, you can start making changes to the code.

- Create a New Branch:** Before making changes, it's recommended to create a new branch. This ensures your changes don't interfere with the main codebase until you're ready to merge.

```
git checkout -b feature/your-feature # Create and switch to a new branch
anant@Anant MINGW64 /d/dev/prac/dev/dev (master)
$ git checkout -b feature/your-feature
Switched to a new branch 'feature/your-feature'
```

- Make Changes:** Edit files as needed using your preferred editor or IDE.
- Stage Changes:** After making changes, you need to stage them before committing.

```
git add . # Stages all modified files
```

- Commit Changes:** Once changes are staged, commit them to your local branch.

```
git commit -m "Add feature X"
anant@Anant MINGW64 /d/dev/prac/dev/dev (feature/your-feature)
$ git add .

anant@Anant MINGW64 /d/dev/prac/dev/dev (feature/your-feature)
$ git commit -m "Add feature X"
On branch feature/your-feature
nothing to commit, working tree clean
```

### Step 4: Push Changes to GitHub

Once you've committed your changes locally, you need to push them to your GitHub repository.

- Push Your Changes** to the remote repository:

```
git push origin feature/your-feature # Push the feature branch to GitHub
```

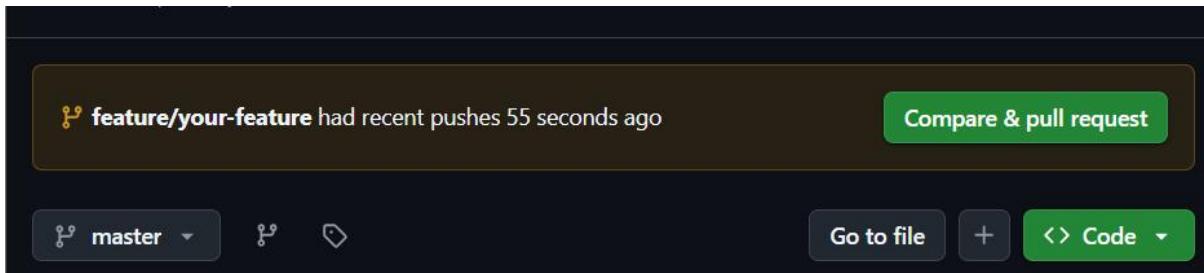
This uploads your local changes to your GitHub repository.

```
anant@Anant MINGW64 /d/dev/prac/dev/dev (feature/your-feature)
$ git push origin feature/your-feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/your-feature' on GitHub by visiting:
remote: https://github.com/AnantKumbhar/dev/pull/new/feature/your-feature
remote:
To https://github.com/AnantKumbhar/dev.git
 * [new branch] feature/your-feature -> feature/your-feature
```

## Step 5: Create a Pull Request (PR)

To contribute your changes back to the original repository, you'll need to open a pull request (PR).

1. **Go to the GitHub Repository:** Visit the repository where you want to make the changes.
2. **Create a Pull Request:** On GitHub, you'll see an option to compare your branch (e.g., `feature/your-feature`) with the main branch of the repository (e.g., `main`). Click **New Pull Request**.



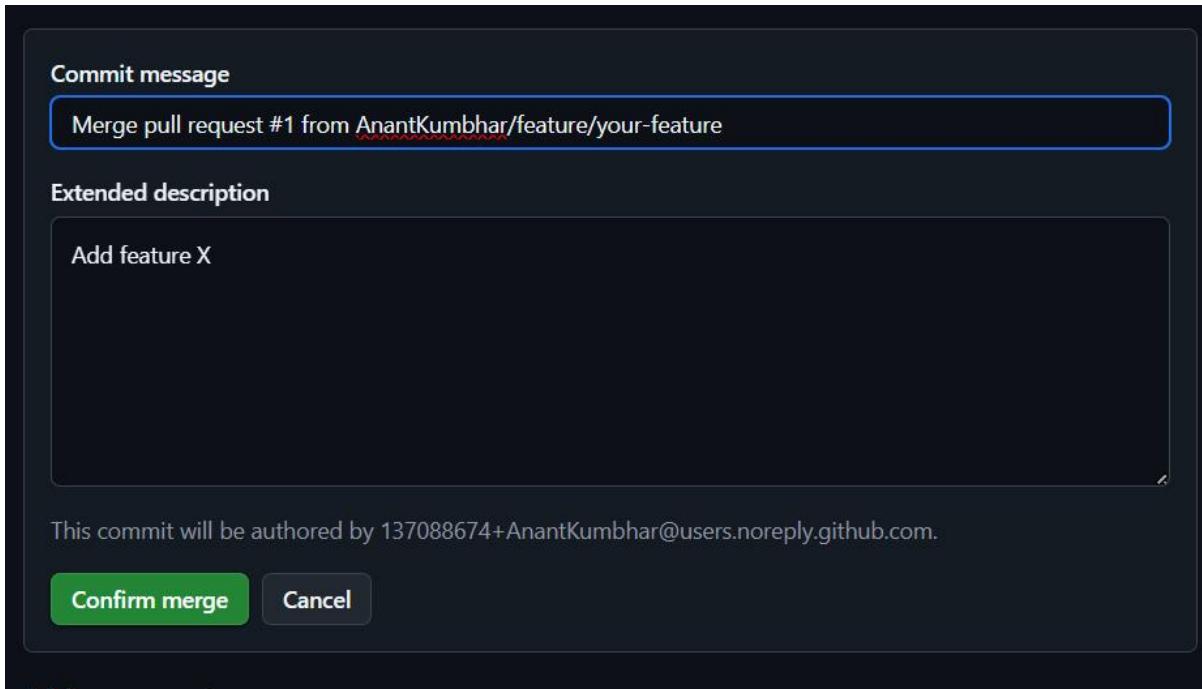
3. **Fill Out the Pull Request Form:**
  - o Add a **title** and **description** explaining the changes you've made.
  - o Review your changes.
  - o Click **Create Pull Request**.

The screenshot shows the GitHub interface for creating a pull request. At the top, there are navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and more. The 'Code' link is highlighted with an orange underline. On the right side of the header are search, file, and profile icons.

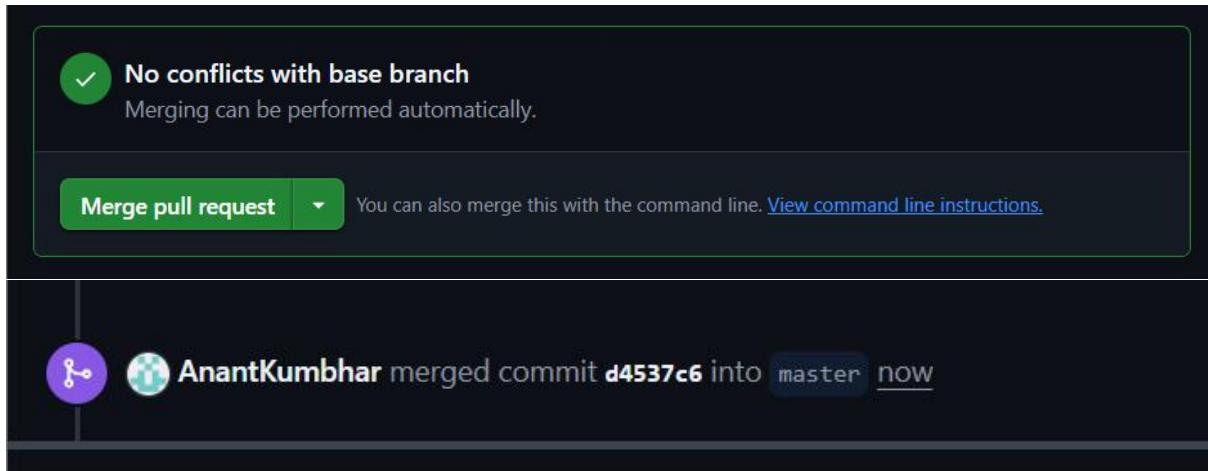
The main section is titled "Open a pull request". It provides instructions: "Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. [Learn more about diff comparisons here.](#)" Below this, there are dropdown menus for "base: master" and "compare: feature/your-feature". A green checkmark indicates "Able to merge. These branches can be automatically merged."

Below the dropdowns, there is a title input field containing "Add feature X" and a rich text editor toolbar labeled "Write" and "Preview". The rich text editor has various icons for bold, italic, underline, etc. A preview window shows the text "text has been edited".

4. **Code Review:** The repository maintainer (or other collaborators) will review your changes. They might ask for changes or approve the PR.



5. **Merge the Pull Request:** Once your changes are approved, the maintainer will merge your changes into the main branch of the project.



## Step 6: Sync Your Fork (If Working on a Forked Repo)

If you are working on a forked repository and want to keep your fork in sync with the original repository:

1. **Add the Original Repository as a Remote:** This allows you to fetch updates from the original repository.

```
git remote add upstream https://github.com/owner/original-repository.git
```

Replace `https://github.com/owner/original-repository.git` with the original repository's URL.

## 2. Fetch the Latest Changes from the Original Repository:

```
git fetch upstream # Fetch the changes from the original repo
anant@Anant MINGW64 /d/dev/prac/dev/dev (feature/your-feature)
$ git remote add upstream https://github.com/AnantKumbhar/dev.git

anant@Anant MINGW64 /d/dev/prac/dev/dev (feature/your-feature)
$ git fetch upstream
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 902 bytes | 300.00 KiB/s, done.
From https://github.com/AnantKumbhar/dev
 * [new branch] feature/your-feature -> upstream/feature/your-feature
 * [new branch] master -> upstream/master
 * [new branch] new_branch -> upstream/new_branch
```

## 3. Merge the Latest Changes into Your Local Branch:

```
git checkout main # Switch to your main branch
git merge upstream/main # Merge the latest changes from the original repo

anant@Anant MINGW64 /d/dev/prac/dev/dev (feature/your-feature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

## 4. Push the Changes to Your Fork:

```
git push origin main # Push the updated main branch to your fork
```

```
praja@LAPTOP-CRE8QUL5 MINGW64 /c/MCA/DevOps/DemoRep/demo (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Anil-03/demo.git
 2fb6142..e41642c master -> master
```

## Step 7: Pull Latest Changes from the Original Repository

To keep your local repository up-to-date with the remote repository on GitHub, you can pull the latest changes.

### 1. Switch to Your Local Main Branch:

```
git checkout main
```

## 2. Pull Latest Changes from GitHub:

```
git pull origin main # Pull the latest changes from the remote repository
anant@Anant MINGW64 /d/dev/prac/dev/dev (master)
$ git pull origin master
From https://github.com/AnantKumbhar/dev
 * branch master -> FETCH_HEAD
 745150a..d4537c6 master -> origin/master
Updating 745150a..d4537c6
Fast-forward
 message.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

anant@Anant MINGW64 /d/dev/prac/dev/dev (master)
$ git pull origin master
From https://github.com/AnantKumbhar/dev
 * branch master -> FETCH_HEAD
Already up to date.
```

## Step 8: Collaborate with Other Developers

When collaborating with other developers on GitHub, you'll typically follow these best practices:

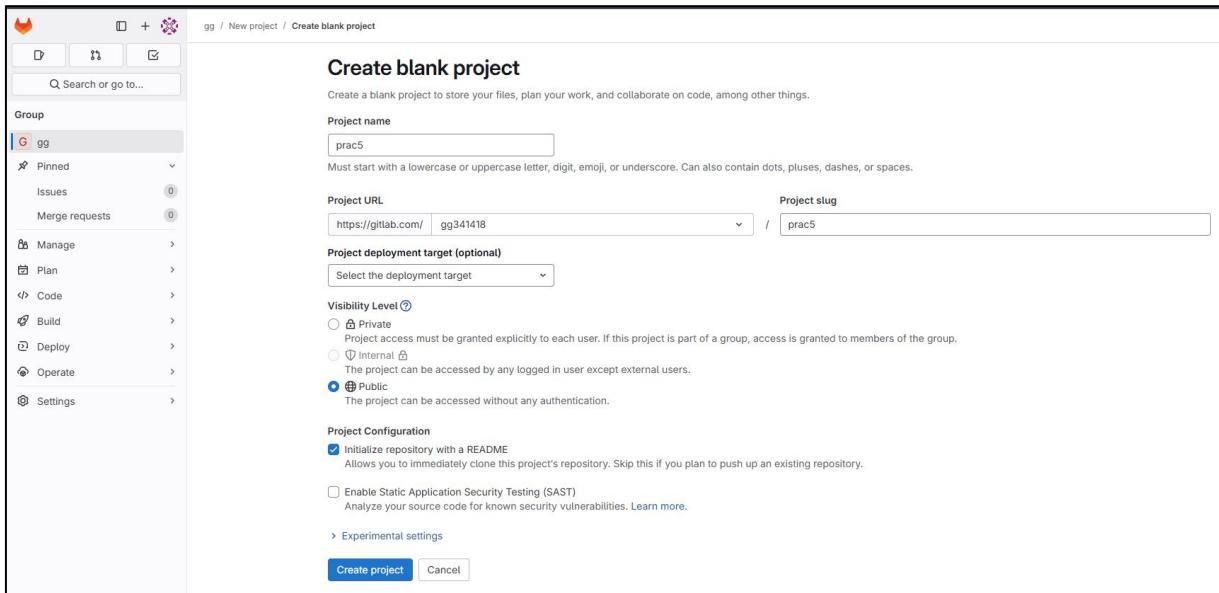
1. **Regularly Pull Latest Changes:** To ensure you're not working on outdated code, frequently pull the latest changes from the main branch (especially before you start working on new features or bug fixes).
2. **Create Feature Branches:** Always create a new branch for each feature or bug fix. This avoids conflicts and keeps the history clean.
3. **Review Pull Requests:** If you're reviewing someone else's PR, ensure you provide feedback and approve it once you're satisfied.
4. **Resolve Merge Conflicts:** If two developers edit the same part of a file, a merge conflict will occur when merging. Resolve these conflicts manually by editing the files and then committing the changes.

# PRACTICAL 5

## AIM: Using Gitlab Web IDE

### Steps:

1. Sign up at <https://gitlab.com>
2. Create a project.
3. Click on Web IDE in your repository.



### 4. Create a file (index.html):

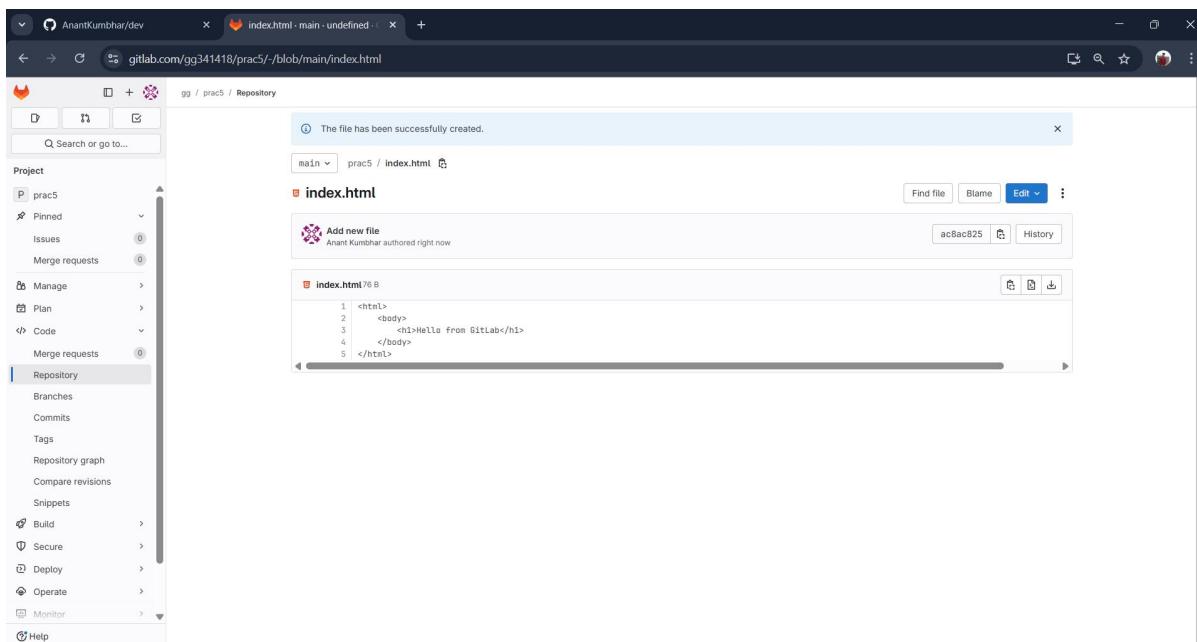
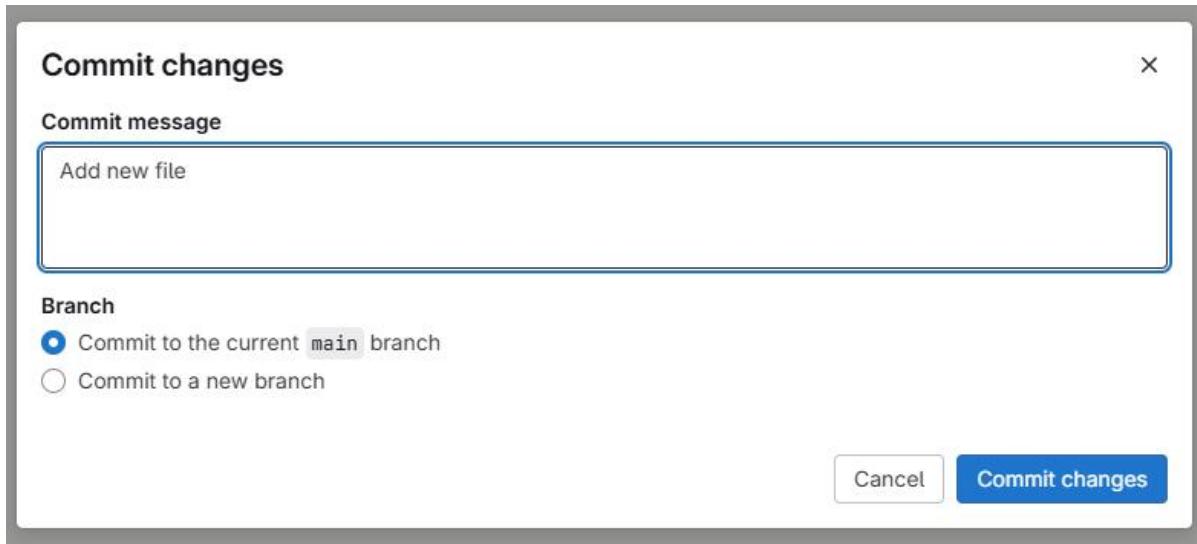
```
<html>
<body>
 <h1>Hello from GitLab</h1>
</body>
</html>
```

**New file**

main	index.html	Cancel	Commit changes
------	------------	--------	----------------

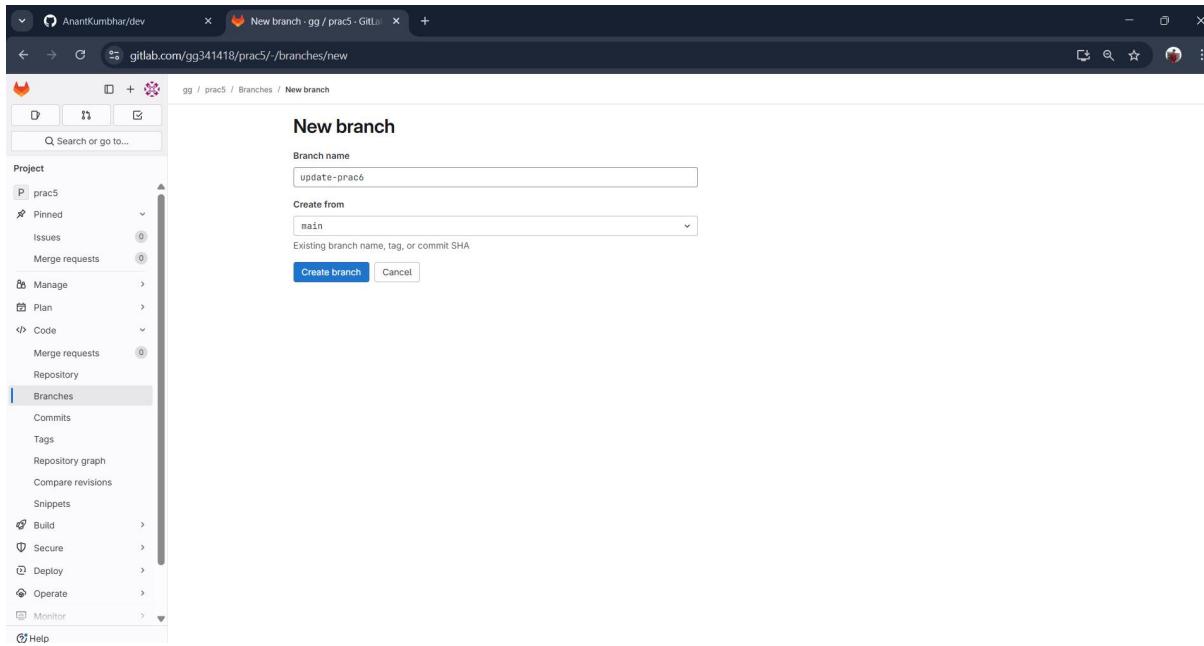
```
1 <html>
2 <body>
3 <h1>Hello from GitLab</h1>
4 </body>
5 </html>
```

No wrap

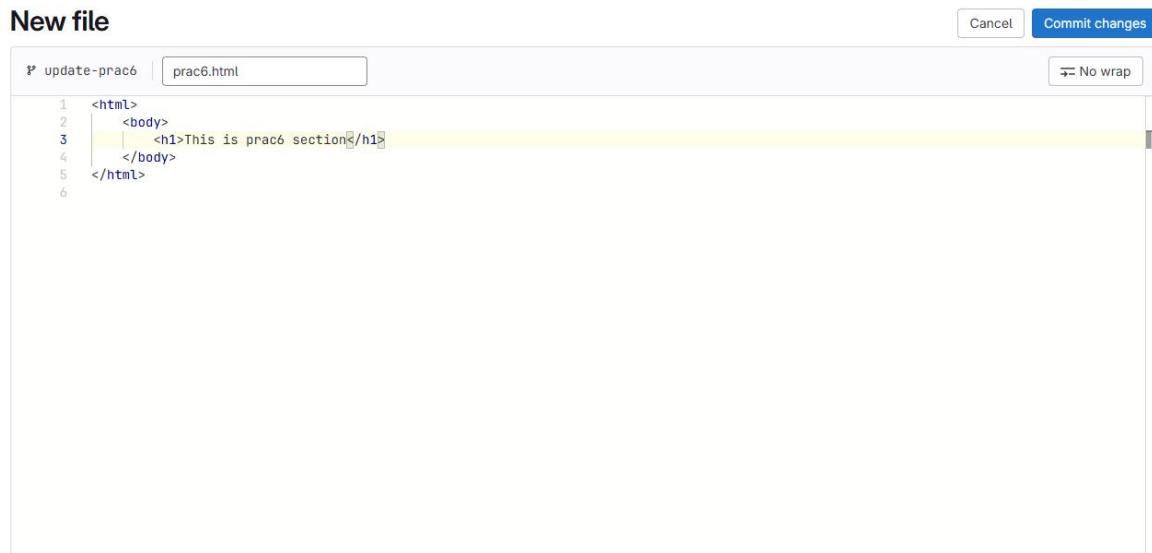
**5. Click Commit and push changes.**

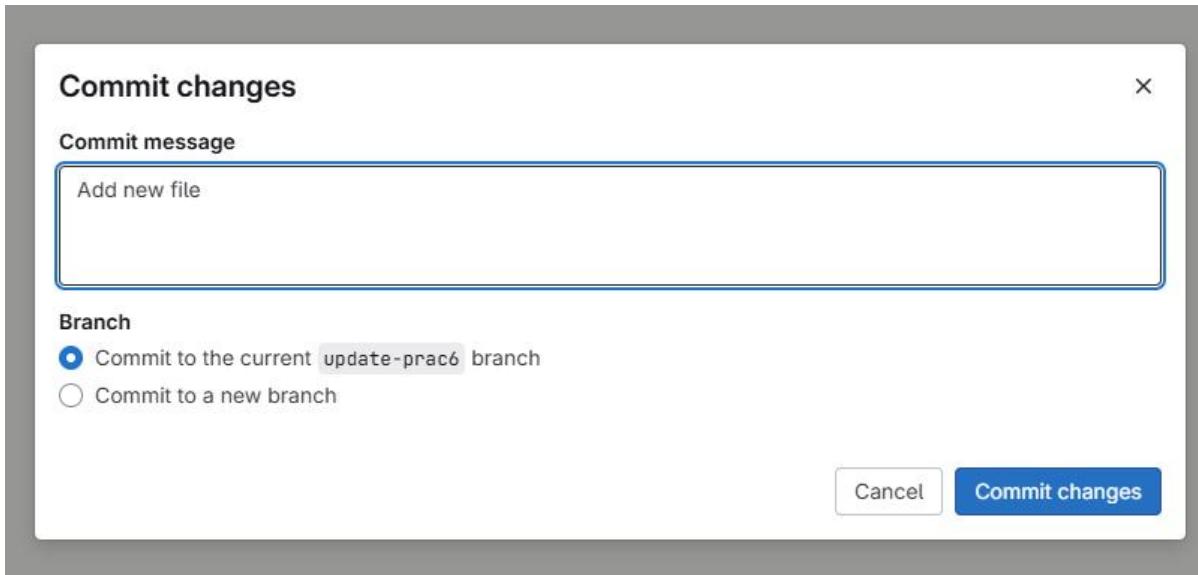
## Performing merge requests using GitLab

### 1. Create a new branch in Web IDE.



### 2. Add /Edit a File and Commit





### 3. Click on merge Request > New Merge request

The file has been successfully created.

You pushed to `update-prac6` 40 minutes ago

**Create merge request**

`update-prac6` `prac5 / prac6.html`

**prac6.html** Find file Blame

### 4. Select source and target branches

AnantKumbhar/dev New merge request - gg / prac5

gitlab.com/gg341418/prac5/-/merge\_requests/new?merge\_request%5Bsource\_branch%5D=update-prac6

**New merge request**

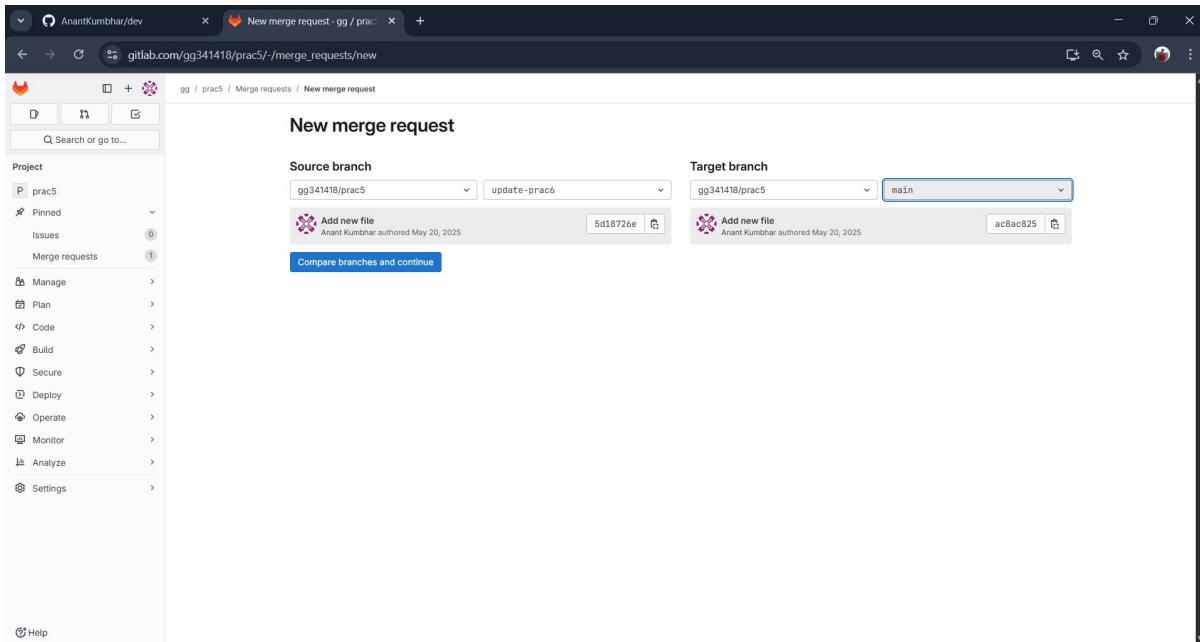
From `update-prac6` into `main` Change branches

**Title (required)**  
Merge  Mark as draft

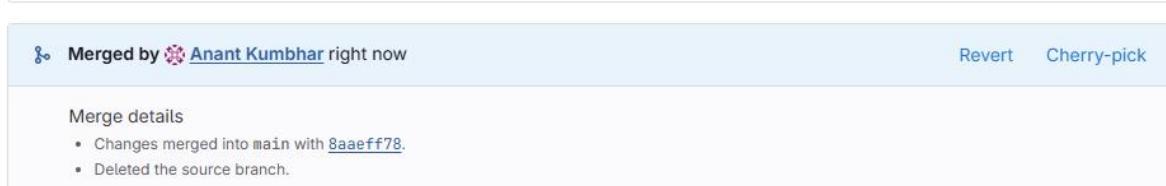
**Description**  
Describe the goal of the changes and what reviewers should be aware of.  
Add description templates to help your contributors to communicate effectively!

**Assignee** Unassigned

**Reviewer** Unassigned



## 5. Submit and merge after review



Merged by  Anant Kumbhar right now

Revert Cherry-pick

Merge details

- Changes merged into `main` with [8aaeff78](#).
- Deleted the source branch.

## Practical 6

**Aim:** Demonstrate the CI/CD workflow in GitLab using .py, .bash, .java file

**Steps:**

### 1. In your repo, create .gitlab-ci.yml:

stages:

- build
- test

build-job:

stage: build

script:

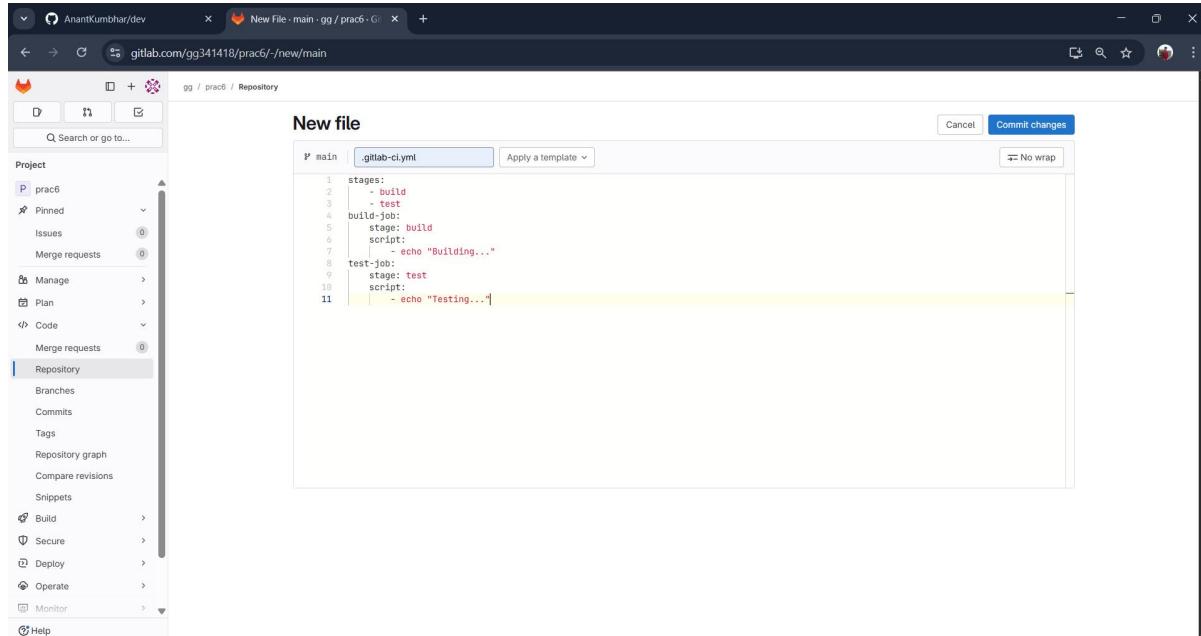
- echo "Building..."

test-job:

stage: test

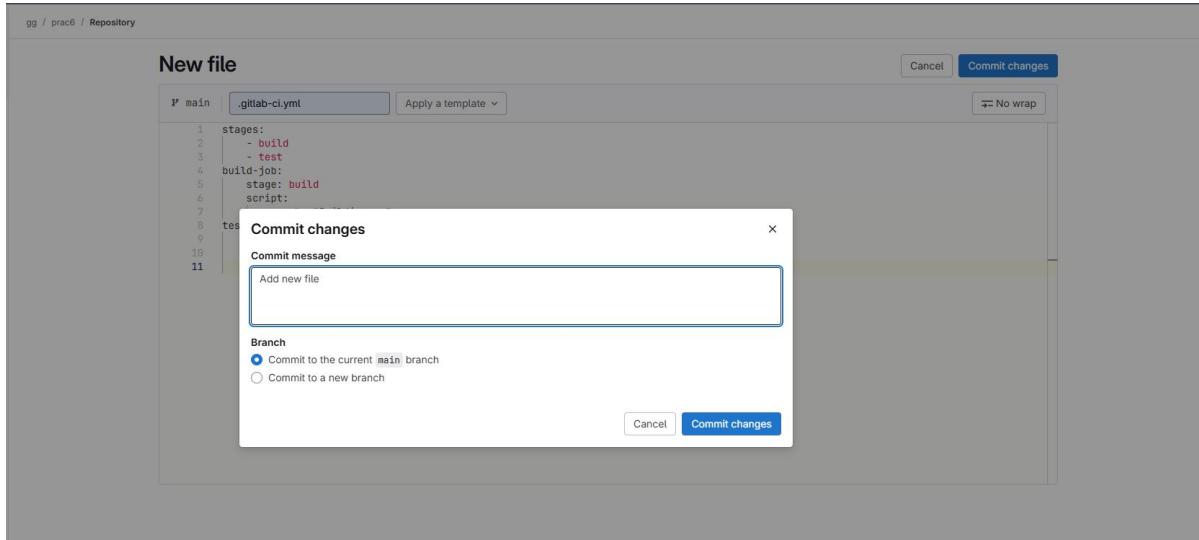
script:

- echo "Testing..."

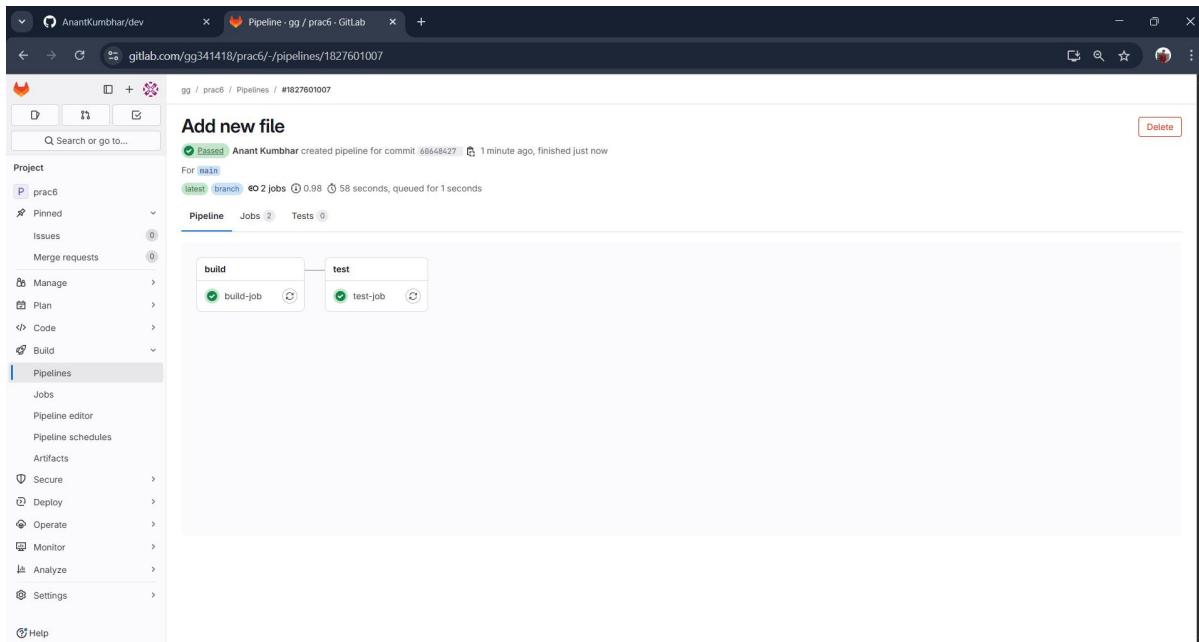


```
main .gitlab-ci.yml
stages:
 - build
 - test
build-job:
 stage: build
 script:
 - echo "Building..."
test-job:
 stage: test
 script:
 - echo "Testing..."
```

## 2. Committe and push



## 3. Go To Bulid Pipeline and View the build/test stages



**build-job**

Passed Started 2 minutes ago by Anant Kumbhar

```

1 Running with gitlab-runner 17.10.0-pre.41.g5c23fd8e (5c23fd8e)
2 on blue-4.saas-linux-small-amd64.runners-manager.gitlab.com/default J2nyww-s, system ID: s_cf1798852952
3 Preparing the "docker+machine" executor
4 Using Docker executor with image ruby:3.1 ...
5 Pulling docker image ruby:3.1 ...
6 Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969006aab67d15c9
2fb930500ff73948803da1330b8a853fecebb5 ...
7 Preparing environment
8 Running on runner-j2nyww-s-project-70015884-concurrent-0 via runner-j2nyww-s-s-l-s-amd64-1747759872-2ac5077e...
9 Getting source from Git repository
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/gg341418/prac6/.git/
12 Created fresh repository.
13 Checking out 60648427 as detached HEAD (ref is main)...
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
16 Executing "step_script" stage of the job script
17 Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969006aab67d15c9
2fb930500ff73948803da1330b8a853fecebb5 ...
18 $ echo "Building...""
19 Building...
20 Cleaning up project directory and file based variables
21 Job succeeded

```

**test-job**

Passed Started 1 minute ago by Anant Kumbhar

```

1 Running with gitlab-runner 17.10.0-pre.41.g5c23fd8e (5c23fd8e)
2 on blue-4.saas-linux-small-amd64.runners-manager.gitlab.com/default -AzERasQ, system ID: s_4cb09cee29e2
3 Preparing the "docker+machine" executor
4 Using Docker executor with image ruby:3.1 ...
5 Pulling docker image ruby:3.1 ...
6 Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969006aab67d15c9
2fb930500ff73948803da1330b8a853fecebb5 ...
7 Preparing environment
8 Running on runner-azerasq-project-70015884-concurrent-0 via runner-azerasq-s-s-l-s-amd64-1747759981-d7593e3a...
9 Getting source from Git repository
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/gg341418/prac6/.git/
12 Created fresh repository.
13 Checking out 60648427 as detached HEAD (ref is main)...
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
16 Executing "step_script" stage of the job script
17 Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969006aab67d15c9
2fb930500ff73948803da1330b8a853fecebb5 ...
18 $ echo "Testing...""
19 Testing...
20 Cleaning up project directory and file based variables
21 Job succeeded

```

**CI/Cd for python****Create script.py**

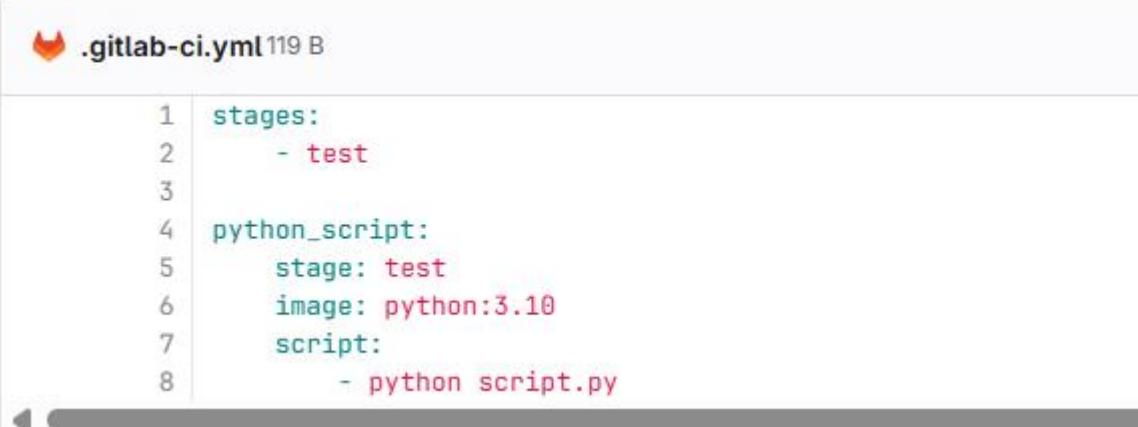
```
print("Hello NMITD!")
```

```

script.py 21 B
1 print("Hello NMITD!")

```

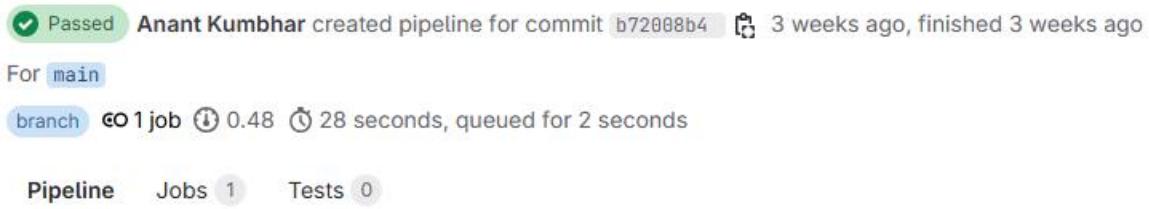
**Create .gitlab-ci.yml**



```
🔥 .gitlab-ci.yml 119 B

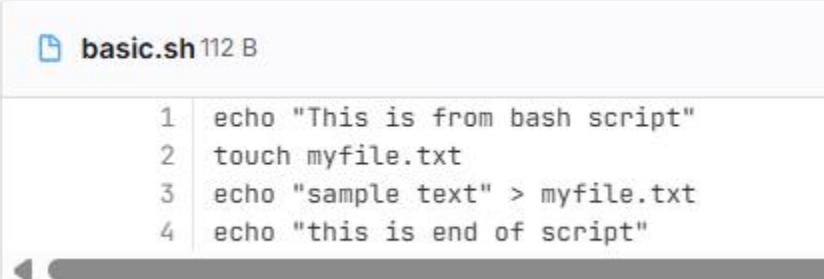
1 stages:
2 - test
3
4 python_script:
5 stage: test
6 image: python:3.10
7 script:
8 - python script.py
```

Commit the changes and build pipeline



CI/Cd for bash

Create basic.sh file



```
📄 basic.sh 112 B

1 echo "This is from bash script"
2 touch myfile.txt
3 echo "sample text" > myfile.txt
4 echo "this is end of script"
```

Create .gitlab-ci.yml

 .gitlab-ci.yml 92 B

```
1 stages:
2 - build
3 bash_execute:
4 stage: build
5 script:
6 - bash basic.sh
```

CI/Cd for Java

Create demo.java

 demo.java 151 B

```
1 class demo{
2 public static void main(String a[]){
3 System.out.println("Hello GG");
4 System.out.println("gg guys helloo");
5 }
6 }
```

Create .gitlab-ci.yml

```
🔥 .gitlab-ci.yml 408 B

1 stages:
2 - build
3 - test
4
5 before_script:
6 - apt-get update && apt-get install -y openjdk-17-jdk
7
8 build:
9 stage: build
10 script:
11 - javac demo.java
12 - ls -ls
13
14 artifacts:
15 paths:
16 - demo.class
17 only:
18 - main
19
20 test:
21 stage: test
22 when: manual
23 script:
24 - ls -l
25 - java demo
26 only:
27 - main
28
29
```

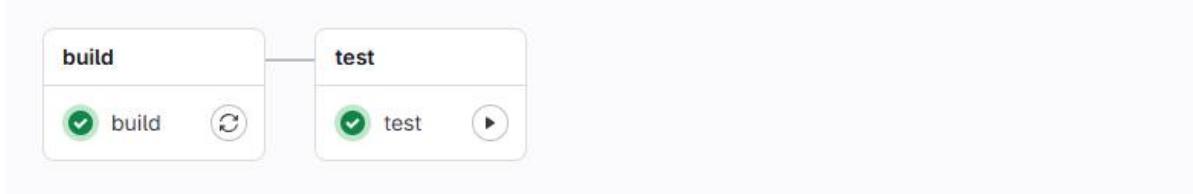
## Commit and build pipeline

Passed Anant Kumbhar created pipeline for commit 660d33ad 3 weeks ago, finished 3 weeks ago

For main

latest branch CI 2 jobs 1.89 1 minute 53 seconds, queued for 1 seconds

Pipeline Jobs 2 Tests 0

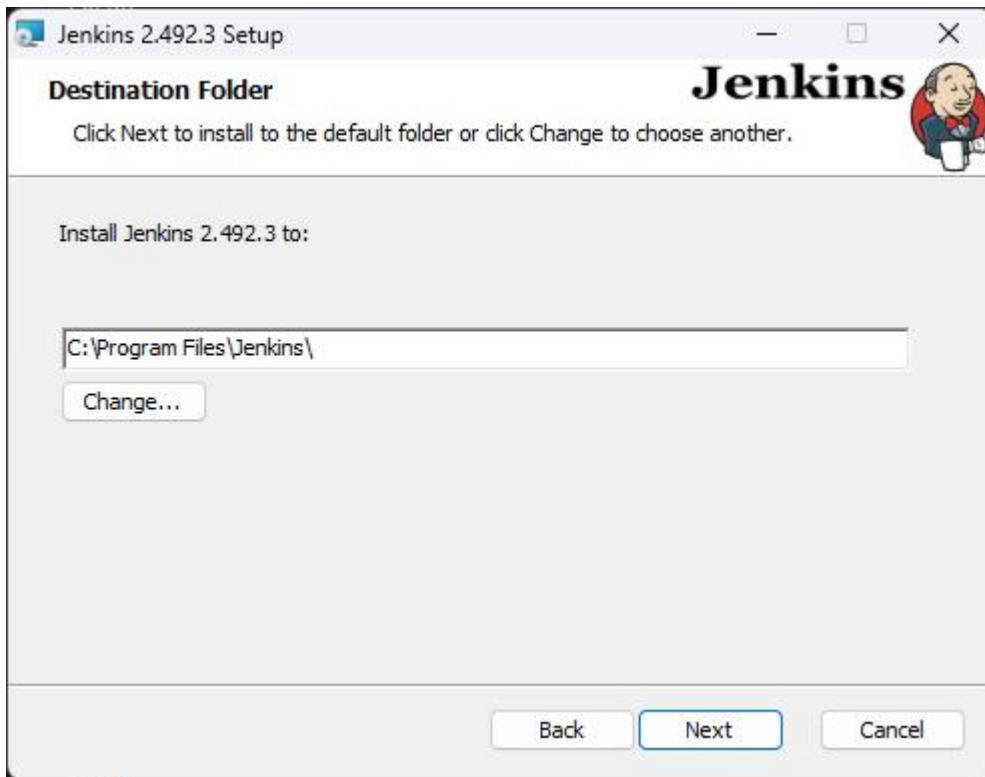


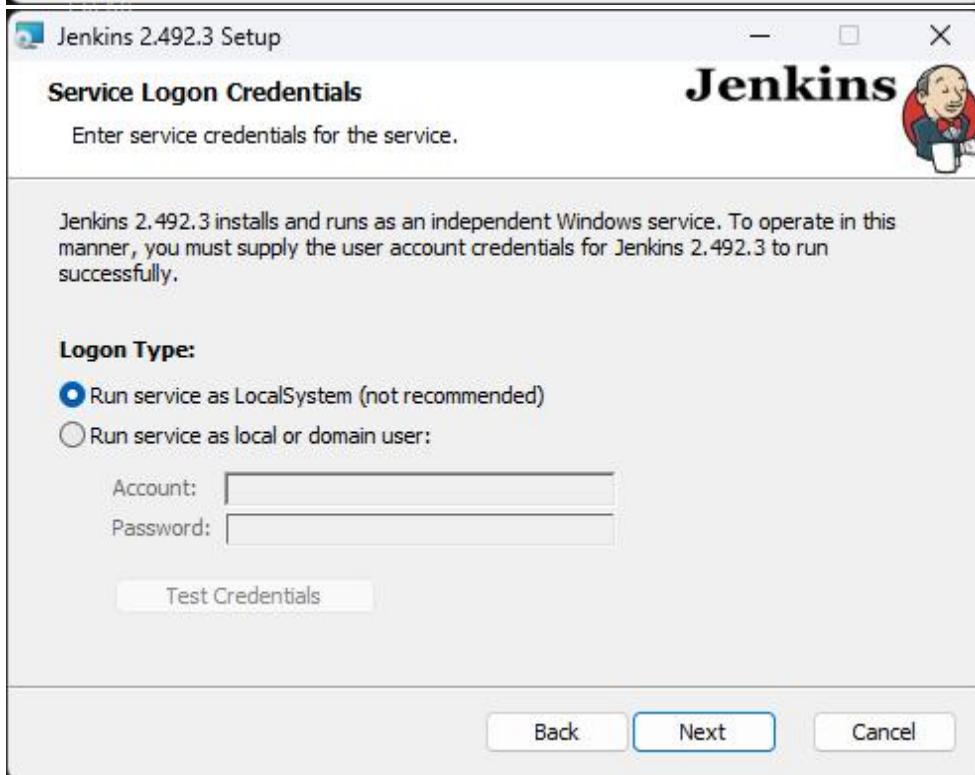
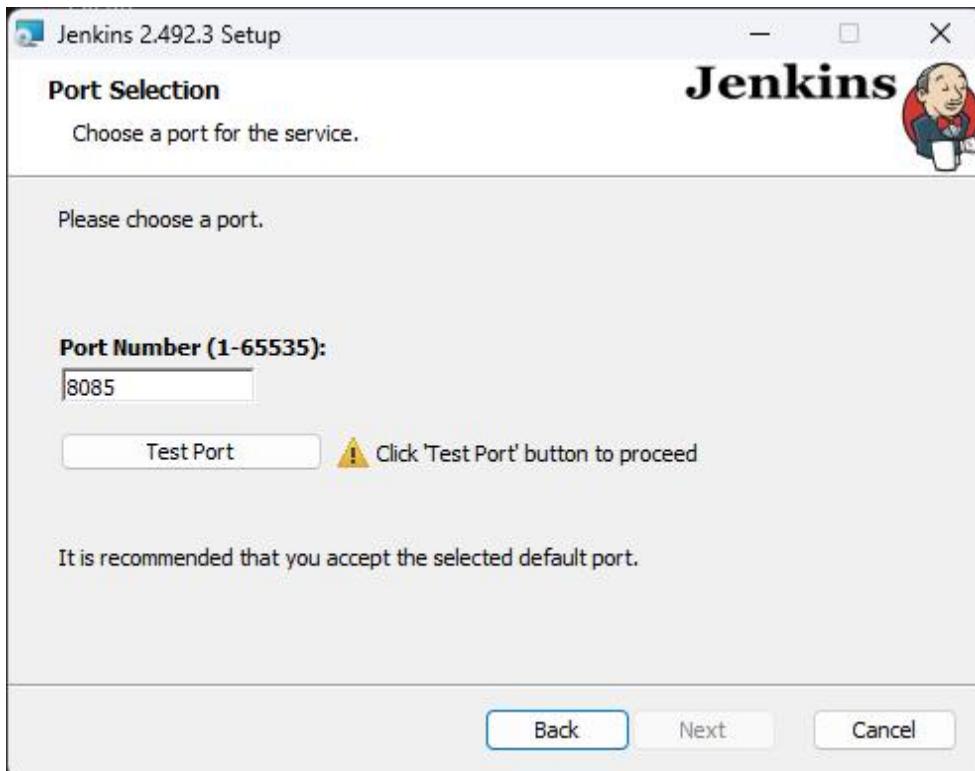
```
812 update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/java to provide java (java) in auto mode
813 $ ls -l
814 total 16
815 -rw-rw-rw- 1 root root 6121 Apr 28 05:25 README.md
816 -rw-r--r-- 1 root root 442 Apr 28 05:24 demo.class
817 -rw-rw-rw- 1 root root 151 Apr 28 05:25 demo.java
818 $ java demo
819 Hello GG
820 gg guys helloo
✓ 821 Cleaning up project directory and file based variables
822 Job succeeded
```

## Practical 7

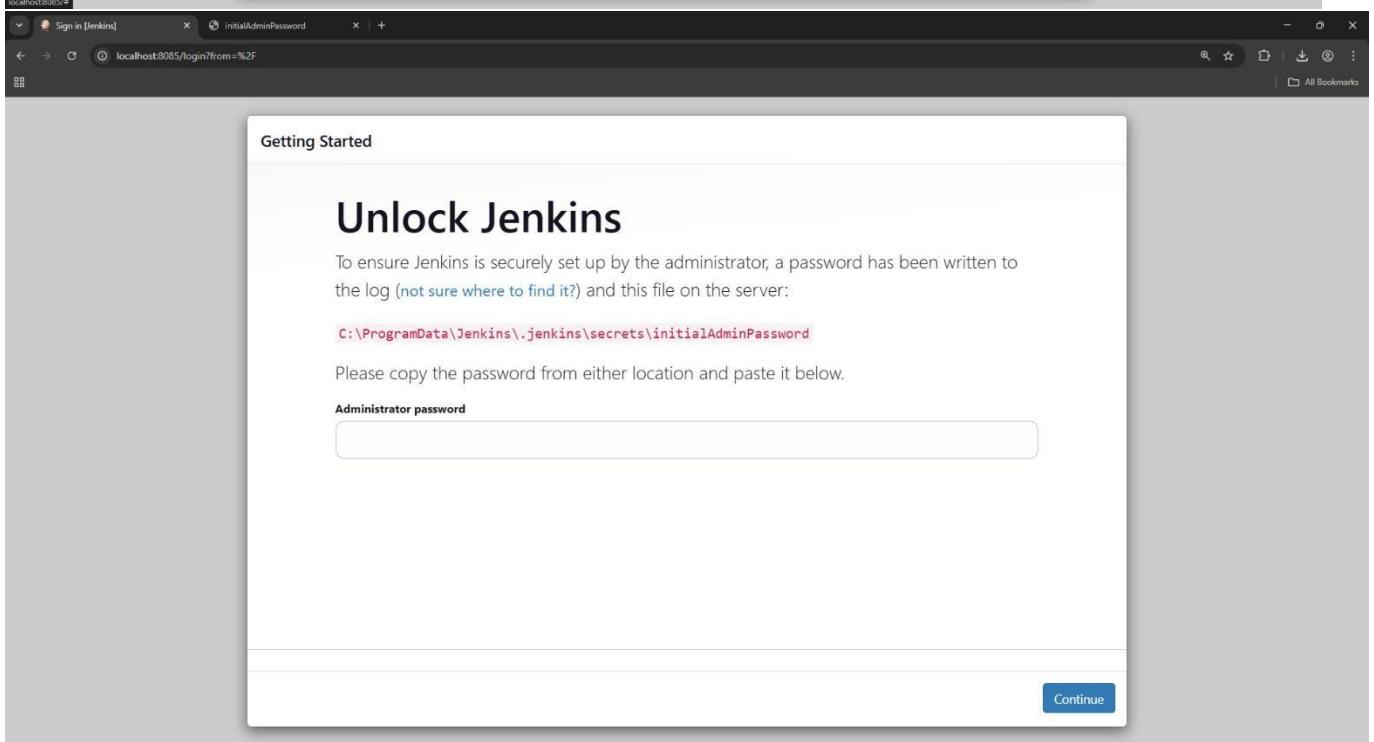
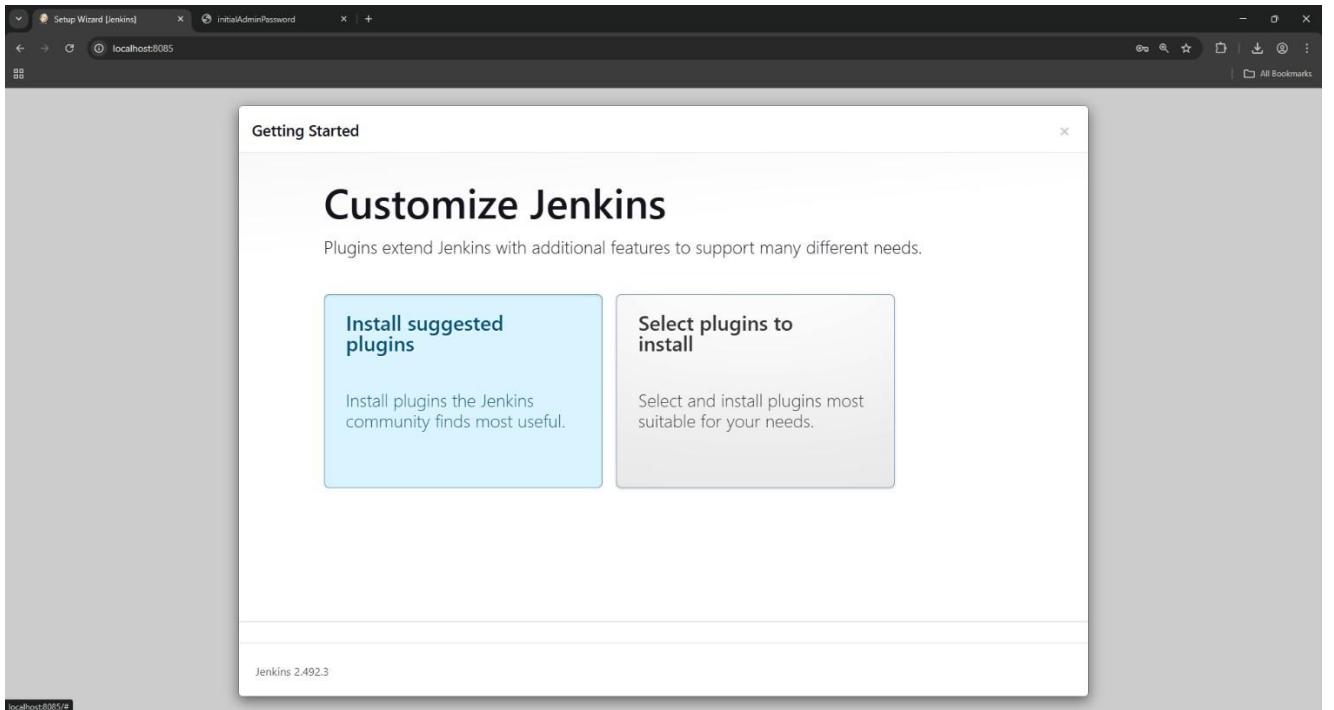
Aim : demonstrate settings jenkins CI/CD pipeline.











## Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** Jackson 2 API ** Commons-text API ** Pipeline: Supporting APIs ** Plugin Utilities API ** Font Awesome API ** Bootstrap 5 API ** JQuery3 API ** ECharts API ** Display URL API ** Checks API ** JUnit ** Matrix Project ** Resource Disposer
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle	Workspace Cleanup Ant ** OkHttp ** Durable Task ** Pipeline: Nodes and Processes ** Pipeline: SCM Step ** Pipeline: Groovy ** Pipeline: Job ** Jakarta Activation API ** Jakarta Mail API ** Apache HttpComponents Client 4.x API ** Instance Identity
✗ Pipeline	✗ GitHub Branch Source	✗ Pipeline: GitHub Groovy Libraries	✗ Pipeline Graph View	Mailer ** Pipeline: Basic Steps Gradle ** Pipeline: Milestone Step ** - required dependency
✗ Git	✗ SSH Build Agents	✗ Matrix Authorization Strategy	✗ PAM Authentication	
✗ LDAP	✗ Email Extension	✓ Mailer	✗ Dark Theme	

Jenkins 2.504.1

Getting Started

# Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD\_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.504.1

Not now Save and Finish

**Getting Started**

# Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.504.1

localhost:8085/view/all/newJob

**Jenkins**

Dashboard > All > New Item

### New Item

Enter an item name  
prac7pipeline

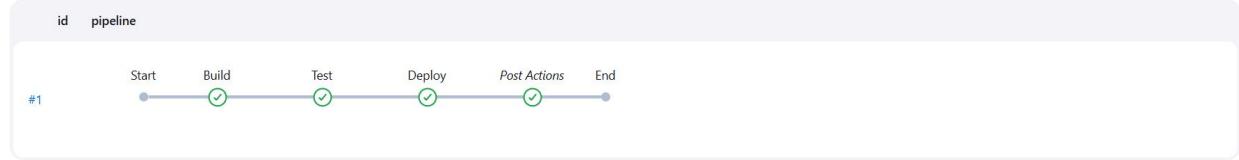
Select an item type

- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**  
Creates a set of multibranch project subfolders by scanning for repositories.

[OK](#)

The screenshot shows the Jenkins pipeline interface for a job named "Build Demopipeline". At the top, there is a navigation bar with the Jenkins logo, a search icon, a shield icon with a red '1', an "admin" dropdown, and a "log out" link. Below the navigation bar, the breadcrumb navigation shows "Dashboard > Demopipeline > Stages". The main title "Build Demopipeline" is centered above a pipeline diagram. The pipeline diagram is titled "id pipeline" and has an ID "#1". It consists of six stages connected by arrows: "Start" (grey dot), "Build" (green circle with checkmark), "Test" (green circle with checkmark), "Deploy" (green circle with checkmark), "Post Actions" (green circle with checkmark), and "End" (grey dot). To the right of the pipeline diagram, there are two buttons: a green "Build" button with a play icon and a "Configure" button.

### Build Demopipeline



Jenkins 2.492.3

## Practical 8

**Aim : Demonstrate Setting up of a CI/CD pipeline to build add deploy a web application to a local HTTP server**

### Index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="Cookies.jsp" method="get">
Name:<input type="text" name="user">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

### Cookies.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

```
pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<%

String username=request.getParameter("user");

Cookie[] cookies=request.getCookies();

int visitCount=0;

boolean userExist=false;

if(cookies!=null){

 for(Cookie cookie:cookies){ if(cookie.getName().equals("visitCount")){

 visitCount=Integer.parseInt(cookie.getValue());

 }

 if(cookie.getName().equals("username")){
 userExist=true;
 }
}

}

}
```

```
visitCount++;

Cookie visitcookie=new Cookie("visitCount",String.valueOf(visitCount));
visitcookie.setMaxAge(60*60*24);

response.addCookie(visitcookie);

if(!userExist&&username!=null){

 Cookie usercookie=new Cookie("username",username);
 usercookie.setMaxAge(60*60*24);
 response.addCookie(usercookie);

}

%>

<p>Hello <%=username!=null? username:"Guest" %> You have hit the Page <%=visitCount %>
times</p>

<a href="Cookies.jsp?user=<%= username %>">Hit Again

</body>

</html>
```

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a project structure for 'Tomcat v10.1 Server at localhost'. A file named 'Cookies.jsp' is selected in the 'src' folder. The code editor window contains the following JSP code:

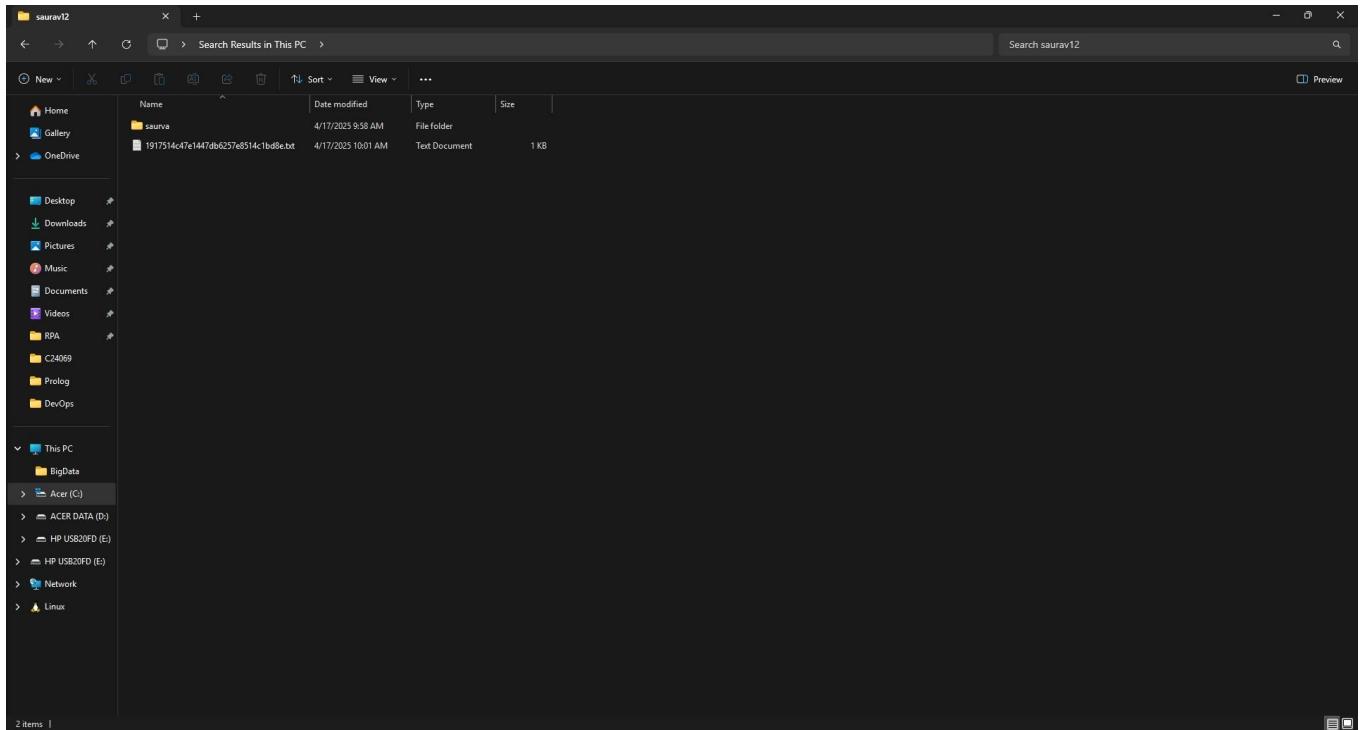
```

<%@ page language="java" contentType="text/html; charset=UTF-8"
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%>
11 String username=request.getParameter("user");
12 Cookie[] userCookies=request.getCookies();
13 int visitCount=0;
14 boolean userExist=false;
15 if(cookies!=null){
16
17 for(Cookie cookie:cookies){
18 if(cookie.getName().equals("visitCount")){
19 visitCount=Integer.parseInt(cookie.getValue());
20 }
21 if(cookie.getName().equals("username")){
22 userExist=true;
23 }
24 }
25 }
26 visitCount++;
27 Cookie visitCookie=new Cookie("visitCount",String.valueOf(visitCount));
28 visitCookie.setMaxAge(60*60*24);
29 response.addCookie(visitCookie);
30 if(userExist&&username!=null){
31 Cookie userCookie=new Cookie("username",username);
32 userCookie.setMaxAge(60*60*24);
33 response.addCookie(userCookie);
34 }
35
36
37 <p>Hello <%=username==null? "Guest" : username %> You have hit the Page <%=visitCount %> times</p>
38 <a href="Cookies.jsp?user=<%=username %>">Hit Again
39 </body>
40 </html>

```

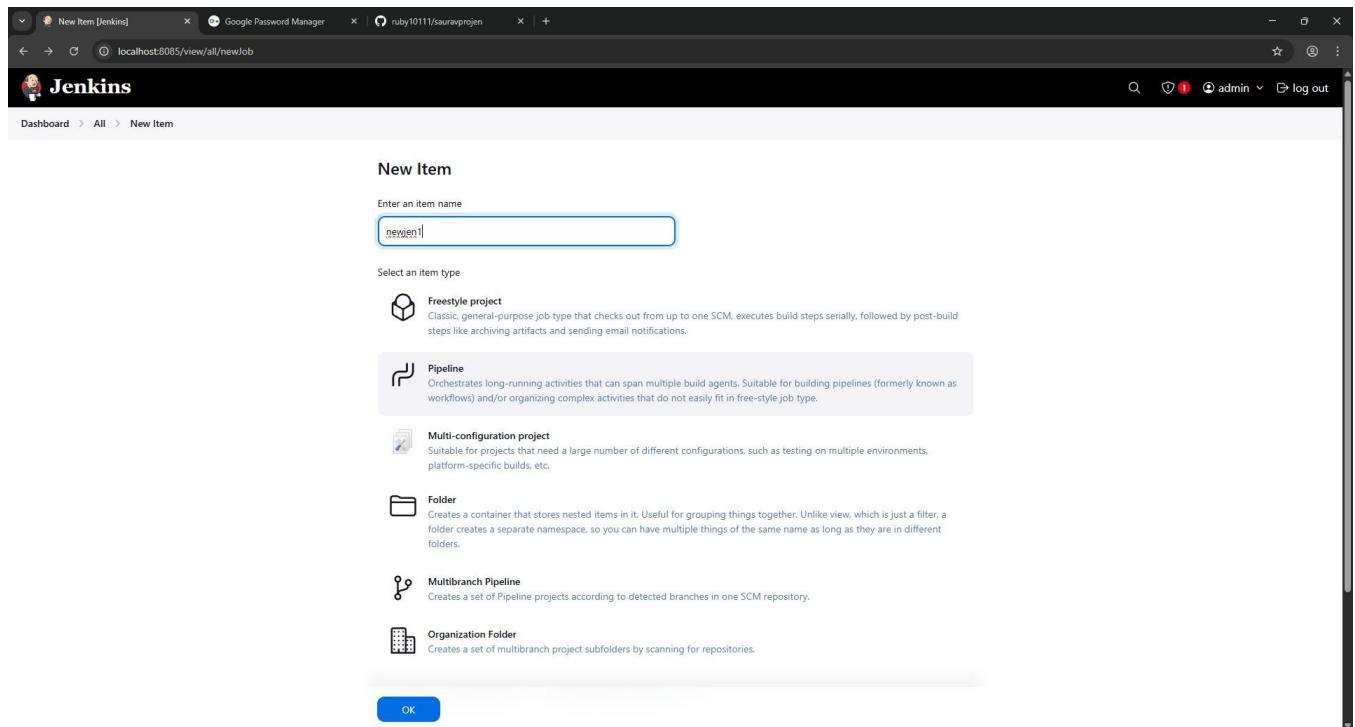
The right side of the interface shows the 'Welcome' screen with various links for getting started with Eclipse, such as 'Review IDE configuration settings', 'Create a Hello World application', 'Create a new Java project', 'Checkout projects from Git', and 'Import existing projects'. A 'Snipping Tool' window is also visible in the bottom right corner.

## Push project on github



A screenshot of a GitHub repository page for 'sauravprojen'. The repository is public and contains one commit by 'ruby10111' and one commit by 'saurva'. The page includes sections for About, Releases, Packages, Languages, and Suggested workflows. It also features a 'Add a README' button.

## Create a pipeline:



```

pipeline{ a
 gent any
 stages{
 stage('Checkout
 Code'){ steps{
 script{
 git branch: 'master',url:'https://github.com/admin111/devopsprojen'
 }
 }
 }
 stage('Verify
 Files'){ steps{
 bat 'dir /S /B'
 }
}

```

```
stage('Deploy'){
 steps{
 script{
 def srcPath="admin/src/main/webapp"
 def destPath="C:\\Program Files\\Apache Software Foundation\\Tomcat
10.1\\webapps\\NewFile"
 if(fileExists(srcPath)){
 bat"xcopy /E /I \"${srcPath}\" \"${destPath}\\\""
 }
 else{
 error "Source directory ${srcPath} does not exists"
 }
 }
 }
}
```

# DES's NMITD

MCA SEM II 2024-25

DEVOPS LAB

ROLL NO: C24049

The screenshot shows the Jenkins configuration interface for a job named 'newjen1'. The 'Triggers' tab is selected. Under the 'Pipeline' section, it says 'Define your Pipeline using Groovy directly or pull it from source control.' A dropdown menu shows 'Pipeline script'. Below is a code editor containing the following Groovy script:

```
script{
 def srcPath="saurva/src/main/webapp"
 def destPath="c:\\Program Files\\Apache Software Foundation\\Tomcat 10.1\\webapps\\NewFile"
 if(fileExists(srcPath)){
 bat"xcopy /E /I \\${srcPath}* \\${destPath}*"
 }
 else{
 error "Source directory ${srcPath} does not exists"
 }
}
```

Use Groovy Sandbox

**Save** **Apply**

The screenshot shows the Jenkins build details for build #9, which was triggered on April 17, 2025, at 10:54:49 AM. The build was started by user 'admin'. It took 2.6 seconds and completed 1 minute 12 seconds ago. The build status is green. The left sidebar shows various build-related options like Status, Changes, Console Output, and Git Build Data. The right panel displays the build summary and specific details for this run.

**Status** #9 (Apr 17, 2025, 10:54:49 AM)

Started by user admin  
This run spent:  
• 7 ms waiting:  
• 2.6 sec build duration:  
• 2.6 sec total from scheduled to completion.

**git**  
Revision: f719956b127acf4ee9bd0f09333775eab88b0ec42  
Repository: https://github.com/ruby10111/sauravprojen  
refs/remotes/origin/master

</> No changes.

**Changes** **Console Output** **Edit Build Information** **Delete build #9\*** **Timings** **Git Build Data** **Pipeline Overview** **Pipeline Console** **Restart from Stage** **Replay** **Pipeline Steps** **Workspaces** **Previous Build**

# DES's NMITD

MCA SEM II 2024-25

DEVOPS LAB

ROLL NO: C24049

The screenshot shows the Jenkins dashboard at [localhost:8085](http://localhost:8085). The main content area displays a table of pipelines. The table has columns for Status (S), Work Queue (W), Name (l), Last Success, Last Failure, and Last Duration. One pipeline, 'Demopipeline', is listed with a green checkmark icon, a yellow sun icon, and the name 'Demopipeline'. Its last success was 7 days 22 hr ago, it has N/A last failure, and its last duration was 1.6 sec. Below the table, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2). On the right side of the dashboard, there are links for 'Add description', 'REST API', and 'Jenkins 2.492.3'.

S	W	Name (l)	Last Success	Last Failure	Last Duration
		Demopipeline	7 days 22 hr <a href="#">#2</a>	N/A	1.6 sec

Build Queue: No builds in the queue.

Build Executor Status: 0/2

Icon: S M L

Add description REST API Jenkins 2.492.3

## Practical 9

### Aim: demonstrate basic Docker commands

1. Check Docker version

```
docker --version
```

```
ubuntu@ubuntu:~$ docker --version
Docker version 28.1.1, build 4eba377
```

2. Pull a Docker image from Docker Hub

```
docker pull nginx
```

```
ubuntu@ubuntu:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
254e724d7786: Pull complete
913115292750: Pull complete
3e544d53ce49: Pull complete
4f21ed9ac0c0: Pull complete
d38f2ef2d6f2: Pull complete
40a6e9f4e456: Pull complete
d3dc5ec71e9d: Pull complete
Digest: sha256:c15da6c91de8d2f436196f3a768483ad32c258ed4e1beb3d367a27ed67253e66
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

3. List all Docker images

```
docker images
```

```
ubuntu@ubuntu:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest a830707172e8 4 weeks ago 192MB
```

4. Run a container from an image

```
docker run -d -p 8080:80 --name mynginx nginx
```

This will run the Nginx container and map port 80 (inside the container) to port 8080 (on your host).

```
ubuntu@ubuntu:~$ docker run -d -p 8080:80 --name mynginx nginx
c241fdc47993e83fe932231e1ba068b8953126eb87a89916c50ebabdc088254c
```

5. List all running containers

```
docker ps
```

```
ubuntu@ubuntu:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c241fdc47993 nginx "/docker-entrypoint..." 27 seconds ago Up 26 seconds 0.0.0.0:8080->80/tcp mynginx
```

6. Copy content from host to container

```
docker cp index.html mynginx:/usr/share/nginx/html/
```

Replace index.html with your actual file. This copies a file into the running container.

```
ubuntu@ubuntu:~$ docker cp index.html mynginx:/usr/share/nginx/html/
lstat /home/ubuntu/index.html: no such file or directory
```

7. Copy content from container to host

```
docker cp mynginx:/usr/share/nginx/html/index.html .
```

```
ubuntu@ubuntu:~$ docker cp index.html mynginx:/usr/share/nginx/html/
lstat /home/ubuntu/index.html: no such file or directory
```

8. Create and use Docker volume for persistent content

```
docker volume create mydata
```

```
docker run -d -p 8081:80 --name nginx_vol -v mydata:/usr/share/nginx/html nginx
```

Now any data added to the /usr/share/nginx/html inside the container will persist even if the container is removed.

```
ubuntu@ubuntu:~$ docker volume create mydata
mydata
ubuntu@ubuntu:~$ docker run -d -p 8081:80 --name nginx_vol -v mydata:/usr/share/nginx/html nginx
85f2708b8c8ec2c1eba2bb88f10a162feec1faa1ad3f86c2f0e8d0ba32e1090a
```

9. List Docker volumes

```
docker volume ls
```

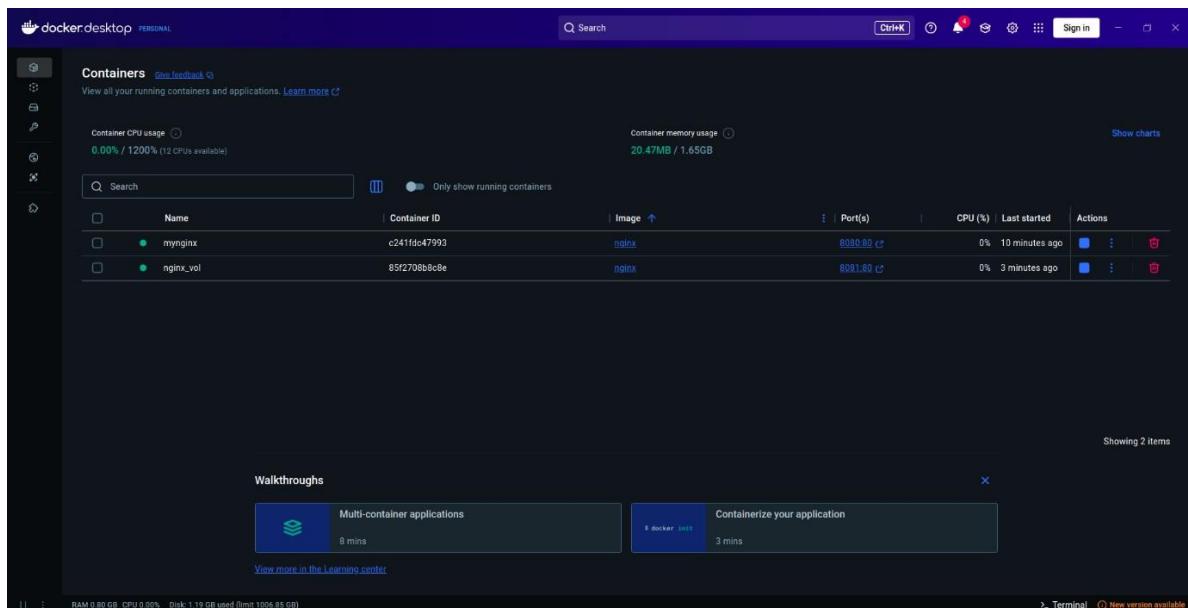
```
ubuntu@ubuntu:~$ docker volume ls
DRIVER VOLUME NAME
local mydata
```

10. Remove a container

```
docker rm -f mynginx
```

Remove an image

```
docker rmi nginx
```



## PRACTICAL-10

**Aim:** Develop a simple containerized application using Docker

Develop a Simple Containerized Application using Docker

1. Index.html



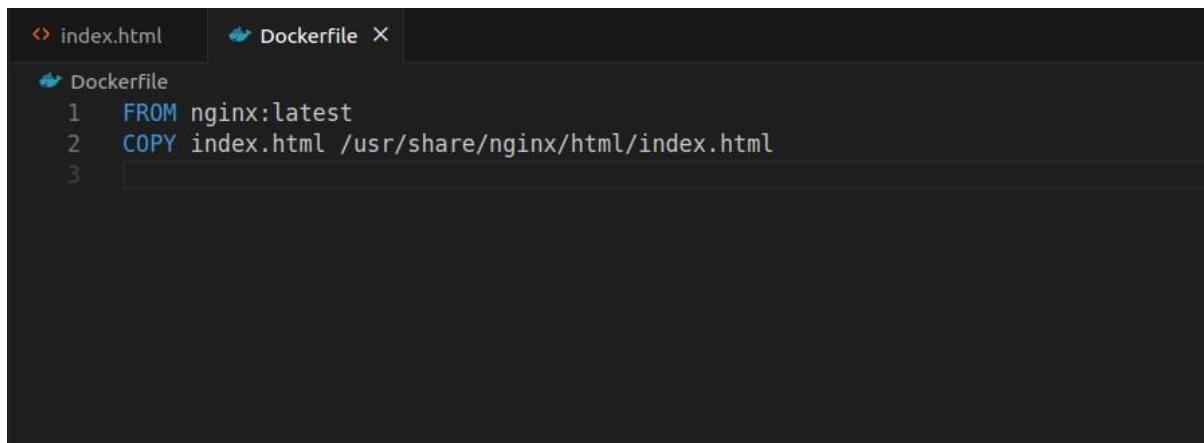
```

index.html X Dockerfile

index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Document</title>
7 </head>
8 <body>
9 <h1>Hello from Docker Container</h1>
10 <h1>Hello From User</h1>
11 </body>
12 </html>

```

2. Dockerfile :-



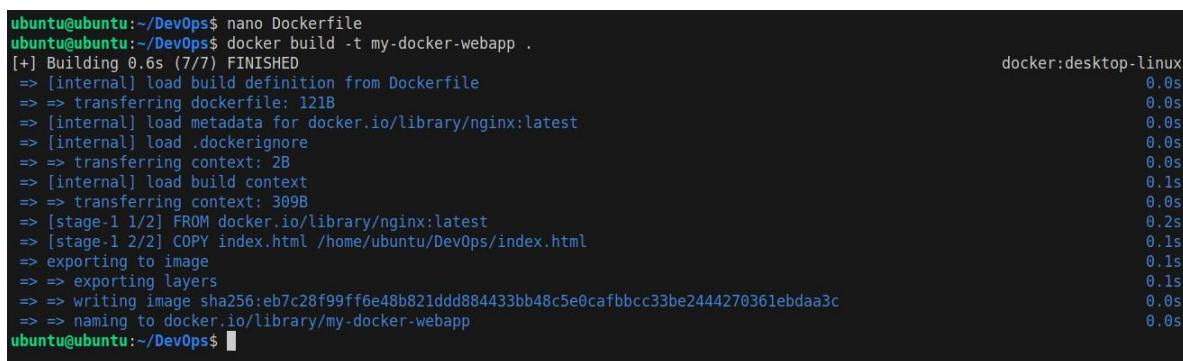
```

index.html Dockerfile X

Dockerfile
1 FROM nginx:latest
2 COPY index.html /usr/share/nginx/html/index.html
3

```

3. docker build -t my-docker-webapp .



```

ubuntu@ubuntu:~/DevOps$ nano Dockerfile
ubuntu@ubuntu:~/DevOps$ docker build -t my-docker-webapp .
[+] Building 0.6s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 121B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 309B
=> [stage-1 1/2] FROM docker.io/library/nginx:latest
=> [stage-1 2/2] COPY index.html /home/ubuntu/DevOps/index.html
=> => exporting to image
=> => exporting layers
=> => writing image sha256:eb7c28f99ff6e48b821ddd884433bb48c5e0cafbbcc33be2444270361ebdaa3c
=> => naming to docker.io/library/my-docker-webapp
ubuntu@ubuntu:~/DevOps$

```

4. docker run -d -p 8080:80 --name webapp-container my-docker-webapp

```
ubuntu@ubuntu:~/DevOps$ docker run -d -p 8080:80 --name webapp-container my-docker-webapp
87758d2c13e4eb227c0bb149148952a661a46b92867ef336a4dd2ad74a993e3f
ubuntu@ubuntu:~/DevOps$
```

5. docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
87758d2c13e4	my-docker-webapp	"/docker-entrypoint..."	38 seconds ago	Up 37 seconds	0.0.0.0:8080->80/tcp	webapp-container
85f2708b8c8e	nginx	"/docker-entrypoint..."	18 minutes ago	Up 18 minutes	0.0.0.0:8081->80/tcp	nginx_vol

6. docker stop webapp-container

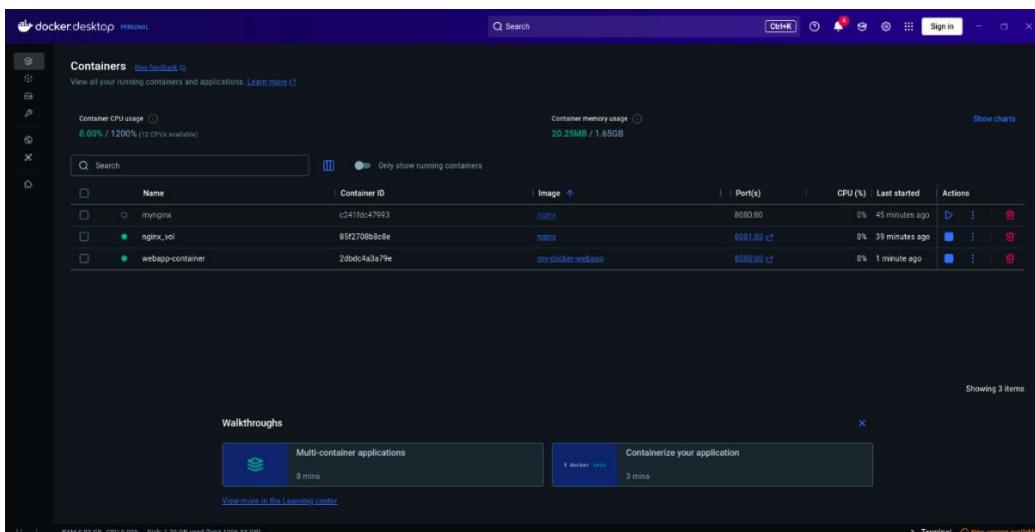
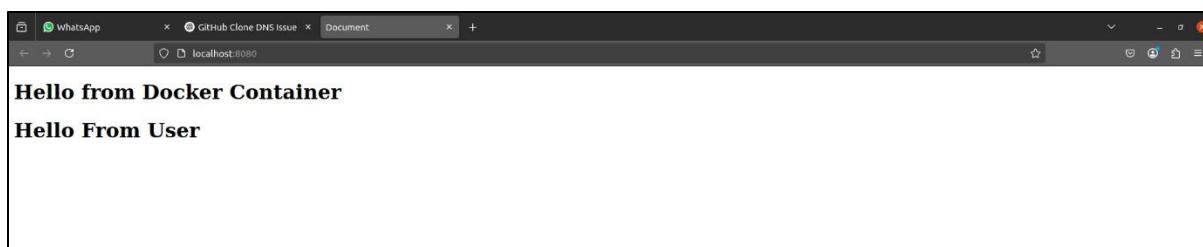
```
ubuntu@ubuntu:~/DevOps$ docker stop webapp-container
webapp-container
```

7. docker rm webapp-container

```
ubuntu@ubuntu:~/DevOps$ docker rm webapp-container
webapp-container
```

8. docker rmi my-docker-webapp

```
ubuntu@ubuntu:~/DevOps$ docker rmi my-docker-webapp
Untagged: my-docker-webapp:latest
Deleted: sha256:eb7c28f99ff6e48b821ddd884433bb48c5e0cafbbcc33be2444270361ebdaa3c
```



## Practical 11

### Aim: Demonstrate add-on ansible commands

#### Step 1: Update your VM

```
ubuntu@ubuntu:~$ sudo apt update && sudo apt upgrade
[sudo] password for ubuntu:
Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 https://packages.microsoft.com/repos/code stable InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:7 http://ppa.launchpad.net/rock-core/qt4/ubuntu focal InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:9 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Hit:10 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:11 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
67 packages can be upgraded. Run 'apt list --upgradable' to see them.
N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository 'https://brave-browser-apt-release.s3.brave.com stable InRelease' doesn't support architecture 'i386'
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
 chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libqt5concurrent5 libqt5opengl5-dev libqt5sql5 libqt5sql5-sqlite libqt5test5 libvulkan-dev libwireshark13
Use 'sudo apt autoremove' to remove them.
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
 vlc-bin vlc-plugin-video-output libavformat58 python2.7-dev libavfilter7
 libavdot4 vlc-plugin-samba python3 libwsresample3 vlc-plugin-gt libzmq5
 python3.7-minimal vlc-plugin-skinn2 vlc-plugin-vte libv4l python3-ipython
 libpython3.7-dev libxv6-graphics-gtk libpostproc5 libbluetooth1 liblcore9
 libavc-bitlibbz2 libzbv10 libavcodec58 vlc libcdts libavutil56 vlc-data
 libavfilter5 libavdevice58 libavmp3r libbsdl2-2.6-0 libmysofa1
 inutils-traceroute vlc-plugin-video-splitter libpython2.7-minimal
 libgraphviz-dev libavcodec libavfilter-base libpython2.7-stdlib traceroute
 libzvbi-common graphviz
Learn more about Ubuntu Pro at http://ubuntu.com/pro
The following NEW packages will be installed:
 linux-headers-5.15.0-139-generic linux-hwe-5.15-headers-5.15.0-139 linux-image-5.15.0-139-generic linux-modules-5.15.0-139-generic linux-modules-extra-5.15.0-139-generic
The following packages will be upgraded:
 code distro-info-data fonts-opensymbol gir1.2-soup-2.4 gnome-shell gnome-shell-common grub-efi-amd64-bin grub-efi-amd64-signed libarchive13 libcryptsetup2 libbjuh-java libjurt-java
 libmysqlclient21 libpoppler-cpp0v5 libpoppler-glib libpoppler97 libraw19 libreoffice-base-core libreoffice-calc libreoffice-common libreoffice-core libreoffice-draw libreoffice-gnome
```

#### Step 2: Install Ansible

```
ubuntu@ubuntu:~$ sudo apt install ansible -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libqt5concurrent5 libqt5opengl5-dev libqt5sql5 libqt5sql5-sqlite libqt5test5 libvulkan-dev libwireshark13
 libwireshark10 libwsutil11 libxext-dev qt5-qmake qt5-qmake-bin qtbase5-dev qtbase5-dev-tools x11proto-xext-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 ieeedata python3-argcomplete python3-crypto python3-dns python3-jinja2 python3-jmespath python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
 python3-requests-kerberos python3-requests-ntlm python3-selinux python3-wlnrm python3-xmldict
Suggested packages:
 cowsay sshpass python-jinja2-doc python-netaddr-docs
The following NEW packages will be installed:
 ansible ieeedata python3-argcomplete python3-crypto python3-dns python3-jinja2 python3-jmespath python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
 python3-requests-kerberos python3-requests-ntlm python3-selinux python3-wlnrm python3-xmldict
0 upgraded, 16 newly installed, 0 to remove and 67 not upgraded.
Need to get 9,726 kB of archives.
After this operation, 90.6 MB of additional disk space will be used.
Ign:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jinja2 all 2.10.1-2ubuntu0.6
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 python3-crypto amd64 2.6.1-1ubuntu2 [237 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-dns python3-all 1.16.0-1ubuntu1 [89.2 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 ieeedata all 20180805.1 [1,589 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-netaddr all 0.7.19-1ubuntu1 [236 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 ansible all 2.9.6+dfsg-1 [5,794 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-argcomplete all 1.8.1-1.3ubuntu1 [27.2 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jmespath all 0.9.4-2ubuntu1 [21.5 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-kerberos amd64 1.1.14-3.1ubuntu1 [22.6 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-libcloud all 2.8.6-1 [1,403 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-ntlm-auth all 1.1.0-1 [19.6 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-ntlm all 1.1.0-1 [6,064 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-selinux amd64 3.0-1ubuntu2 [139 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-wlnrm all 0.12.0-1 [12.6 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 python3-wlnrm all 0.3.0-2 [21.7 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jinja2 all 2.10.1-2ubuntu0.6 [96.3 kB]
Fetched 9,726 kB in 15s (669 kB/s)
Selecting previously unselected package python3-jinja2.
(Reading database ... 212961 files and directories currently installed.)
Preparing to unpack .../00-python3-jinja2_2.10.1-2ubuntu0.6_all.deb ...
Unpacking python3-jinja2 (2.10.1-2ubuntu0.6) ...
Selecting previously unselected package python3-crypto.
```

#### Step 3: Check version:

```
ubuntu@ubuntu:~$ ansible --version
ansible 2.9.6
 config file = /etc/ansible/ansible.cfg
 configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
 ansible python module location = /usr/lib/python3/dist-packages/ansible
 executable location = /usr/bin/ansible
 python version = 3.8.10 (default, Mar 18 2025, 20:04:55) [GCC 9.4.0]
ubuntu@ubuntu:$
```

```
ubuntu@ubuntu:~$ nano host.ini
ubuntu@ubuntu:~$
```

```
GNU nano 4.8
localhost ansible_connection=local
```

### 1. Ping the remote host

**ansible local -i host.ini -m ping**

```
ubuntu@ubuntu:~$ ansible local -i host.ini -m ping
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
localhost | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python"
 },
 "changed": false,
 "ping": "pong"
}
ubuntu@ubuntu:~$
```

### 2. Check uptime

**ansible local -i host.ini -a "uptime"**

```
ubuntu@ubuntu:~$ ansible local -i host.ini -a "uptime"
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
localhost | CHANGED | rc=0 >>
16:31:16 up 2:49, 1 user, load average: 1.08, 0.98, 0.90
ubuntu@ubuntu:~$
```

### 3. Install a package

**ansible local -i host.ini -m apt -a "name=nginx state=present update\_cache=yes" --become**

```
ubuntu@ubuntu:~$ ansible local -i host.ini -m apt -a "name=nginx state=present update_cache=yes" --become
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
localhost | CHANGED => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python"
 },
 "cache_update_time": 1747566323,
 "cache_updated": true,
 "changed": true,
 "stderr": "",
 "stdout_lines": [],
 "stdout": "Reading package lists...\\nBuilding dependency tree...\\nReading state information...\\nThe following packages were automatically installed and are no longer required:\\n chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi\\n libgstreamer-plugins-bad1.0-0 libgbtconcurrent5 libgbtopenpgl5-dev libgbtql5\\n libgbtssql5-sqlite libgbtstest5 libvulkan-dev libwireshark13 libwiretap10\\n libwsutil111 libtxbext-dev qt5-qmake qt5-qmake-bin qtbase5-dev libtxproto-xext-dev\\n Use 'sudo apt autoremove' to remove them.\\n The following additional packages will be installed:\\n libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter\\n libnginx-mod-mail libnginx-mod-stream libnginx-common libnginx-core\\n Suggested packages:\\n fcgiwrap nginx-doc\\n The following NEW packages will be installed:\\n libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter\\n libnginx-mod-mail libnginx-mod-stream libnginx nginx nginx-common\\n nginx-core\\n Upgrading libnginx-mod-http-image-filter from 1.18.0-0ubuntu1.7 to 1.18.0-0ubuntu1.8 (37.8 kB)\\n Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-0ubuntu1.7 [37.8 kB]\\n Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-0ubuntu1.7 [13.1 kB]\\n Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-mail amd64 1.18.0-0ubuntu1.7 [43.0 kB]\\n Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-stream amd64 1.18.0-0ubuntu1.7 [425 kB]\\n Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx amd64 1.18.0-0ubuntu1.7 [425 kB]\\n Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-common amd64 1.18.0-0ubuntu1.7 [425 kB]\\n Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-core amd64 1.18.0-0ubuntu1.7 [425 kB]\\n Reading database ... 5%\\n (Reading database ... 10%\\n (Reading database ... 15%\\n (Reading database ... 20%\\n (Reading database ... 25%\\n (Reading database ... 30%\\n (Reading database ... 35%\\n (Reading database ... 40%\\n (Reading database ... 45%\\n (Reading database ... 50%\\n (Reading database ... 55%\\n (Reading database ... 60%\\n (Reading database ... 65%\\n (Reading database ... 70%\\n (Reading database ... 75%\\n (Reading database ... 80%\\n (Reading database ... 85%\\n (Reading database ... 90%\\n (Reading database ... 95%\\n (Reading database ... 100%\\n (Reading database ... 222461 files and directories currently installed.)\\n Preparing to unpack .../0-nginx-common_1.18.0-0ubuntu1.7_all.deb ...\\n Unpacking nginx-common (1.18.0-0ubuntu1.7) ...\\n Selecting previously unselected package libnginx-mod-http-image-filter.\\n Preparing to unpack .../1-libnginx-mod-http-image-filter_1.18.0-0ubuntu1.7_amd64.deb ...\\n Unpacking libnginx-mod-http-image-filter (1.18.0-0ubuntu1.7) ...\\n Selecting previously unselected package libnginx-mod-mail.\\n Preparing to unpack .../2-libnginx-mod-mail_1.18.0-0ubuntu1.7_amd64.deb ...\\n Unpacking libnginx-mod-mail (1.18.0-0ubuntu1.7) ...\\n Selecting previously unselected package libnginx-mod-stream.\\n Preparing to unpack .../3-libnginx-mod-stream_1.18.0-0ubuntu1.7_amd64.deb ...\\n Unpacking libnginx-mod-stream (1.18.0-0ubuntu1.7) ...\\n Selecting previously unselected package nginx-core.\\n Preparing to unpack .../5-nginx-core_1.18.0-0ubuntu1.7_all.deb ...\\n Unpacking nginx-core (1.18.0-0ubuntu1.7) ...\\n Selecting previously unselected package nginx.\\n Preparing to unpack .../6-nginx_1.18.0-0ubuntu1.7_all.deb ...\\n Unpacking nginx (1.18.0-0ubuntu1.7) ...\\n Setting up nginx-common (1.18.0-0ubuntu1.7) ...\\n Created symlink /etc/systemctl/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.\\n Setting up libnginx-mod-mail (1.18.0-0ubuntu1.7) ...\\n Setting up libnginx-mod-stream (1.18.0-0ubuntu1.7) ...\\n Setting up nginx (1.18.0-0ubuntu1.7) ...\\n Setting up nginx (1.18.0-0ubuntu1.7) ...\\n Processing triggers for man-db (2.9.1-1) ...\\n Processing triggers for ufw (0.36-6ubuntu1.1) ...\\n
```

```
ubuntu@ubuntu:~
```

```

"(Reading database ... 70%",
"(Reading database ... 75%",
"(Reading database ... 80%",
"(Reading database ... 85%",
"(Reading database ... 90%",
"(Reading database ... 95%",
"(Reading database ... 100%",
"(Reading database ... 222461 files and directories currently installed.)",
"Preparing to unpack .../0-nginx-common_1.18.0-0ubuntu1.7_all.deb ...",
"Unpacking nginx-common (1.18.0-0ubuntu1.7) ...",
"Selecting previously unselected package libnginx-mod-http-image-filter.",
"Preparing to unpack .../1-libnginx-mod-http-image-filter_1.18.0-0ubuntu1.7_amd64.deb ...",
"Unpacking libnginx-mod-http-image-filter (1.18.0-0ubuntu1.7) ...",
"Selecting previously unselected package libnginx-mod-http-xslt-filter.",
"Preparing to unpack .../2-libnginx-mod-http-xslt-filter_1.18.0-0ubuntu1.7_amd64.deb ...",
"Unpacking libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.7) ...",
"Selecting previously unselected package libnginx-mod-mail.",
"Preparing to unpack .../3-libnginx-mod-mail_1.18.0-0ubuntu1.7_amd64.deb ...",
"Unpacking libnginx-mod-mail (1.18.0-0ubuntu1.7) ...",
"Selecting previously unselected package libnginx-mod-stream.",
"Preparing to unpack .../4-libnginx-mod-stream_1.18.0-0ubuntu1.7_amd64.deb ...",
"Unpacking libnginx-mod-stream (1.18.0-0ubuntu1.7) ...",
"Selecting previously unselected package nginx-core.",
"Preparing to unpack .../5-nginx-core_1.18.0-0ubuntu1.7_amd64.deb ...",
"Unpacking nginx-core (1.18.0-0ubuntu1.7) ...",
"Selecting previously unselected package nginx.",
"Preparing to unpack .../6-nginx_1.18.0-0ubuntu1.7_all.deb ...",
"Unpacking nginx (1.18.0-0ubuntu1.7) ...",
"Setting up nginx-common (1.18.0-0ubuntu1.7) ...",
"Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service -> /lib/systemd/system/nginx.service.",
"Setting up libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.7) ...",
"Setting up libnginx-mod-mail (1.18.0-0ubuntu1.7) ...",
"Setting up libnginx-mod-http-image-filter (1.18.0-0ubuntu1.7) ...",
"Setting up libnginx-mod-stream (1.18.0-0ubuntu1.7) ...",
"Setting up nginx-core (1.18.0-0ubuntu1.7) ...",
"Setting up nginx (1.18.0-0ubuntu1.7) ...",
"Processing triggers for systemd (245.4-4ubuntu3.24) ...",
"Processing triggers for man-db (2.9.1-1) ...",
"Processing triggers for ufw (0.36-6ubuntu1.1) ..."
]

ubuntu@ubuntu:~$
```

#### 4. Start a service

```
ansible local -i host.ini -m service -a "name=nginx state=started" --become
```

```
ubuntu@ubuntu:~$ ansible local -i host.ini -m service -a "name=nginx state=started" --become
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
localhost | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python"
 },
 "changed": false,
 "name": "nginx",
 "state": "started",
 "status": {
 "ActiveEnterTimestamp": "Sun 2025-05-18 16:35:34 IST",
 "ActiveEnterTimestampMonotonic": "8967169875",
 "ActiveExitTimestampMonotonic": "0",
 "ActiveState": "active",
 "After": "basic.target systemd-journal.socket network.target system.slice sysinit.target",
 "AllowIsolate": "no",
 "AllowedCPUs": "",
 "AllowedMemoryNodes": "",
 "AmbientCapabilities": "",
 "AssertResult": "yes",
 "AssertTimestamp": "Sun 2025-05-18 16:35:34 IST",
 "AssertTimestampMonotonic": "8907060424",
 "Assertion": "multi-user.target shutdown.target",
 "BlockIOAccounting": "no",
 "BlockIOWeight": "[not set]",
 "CPUAcctunting": "no",
 "CPUAffinity": "",
 "CPUAffinityFromLMA": "no",
 "CPUQuotaPerSecUsec": "Infinity",
 "CPUQuotaPeriodUsec": "Infinity",
 "CPUSchedulingPolicy": "0",
 "CPUSchedulingPriority": "0",
 "CPUSchedulingResetOnFork": "no",
 "CPUShares": "[not set]",
 "CPUUsageSec": "[not set]",
 "CPUWeight": "[not set]",
 "CachedDirectoryMode": "0755",
 "CanIsolate": "no",
 "CanReload": "yes",
 "CanStart": "yes",
 ...
 }
}
```

```

 "UnitFileState": "enabled",
 "UtmpMode": "init",
 "WantedBy": "multi-user.target",
 "WatchdogSignal": "6",
 "WatchdogTimestampMonotonic": "0",
 "WatchdogUsec": "0"
}
ubuntu@ubuntu:~$
```

## PRACTICAL-12

Aim: Demonstrate Ansible Playbooks

Install and Start Nginx

**install\_nginx.yml:**

```
- name: Install and start Nginx on web servers
hosts: webservers
```

**become: true tasks:**

```
- name: Install Nginx
apt:
 name: nginx
 state: present
```

**update\_cache: yes**

```
- name: Start Nginx
service:
 name: nginx
 state: started
 enabled: true
```

```
ubuntu@ubuntu:~$ nano install_nginx.yml
```

The screenshot shows a terminal window titled "ubuntu@ubuntu: ~" with the command "nano install\_nginx.yml" entered. The file content is displayed in the editor:

```
GNU nano 4.8
- name: Install and start Nginx on localhost
 hosts: local
 become: yes
 tasks:
 - name: Install nginx
 apt:
 name: nginx
 state: present
 update_cache: yes
 - name: Start nginx
 service:
 name: nginx
 state: started
 enabled: true
```

The terminal window includes standard nano key bindings at the bottom.

**Run the Playbook:****ansible-playbook -i hosts.ini install\_nginx.yml**

```
ubuntu@ubuntu:~$ ansible-playbook -i host.ini install_nginx.yml
PLAY [Install and start Nginx on localhost] ****
TASK [Gathering Facts] ****
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [localhost]
TASK [Install nginx] ****
ok: [localhost]
TASK [Start nginx] ****
ok: [localhost]
PLAY RECAP ****
localhost : ok=3 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

