

PRACTICAL 1

AIM: Demonstrate basic git commands

1. Set-up Git (One-time setup)
 - **git config --global user.name ""** -- Set your name for commits.
 - **git config --global user.email ""**-- Set your email for commits.
2. Start a new project
 - **git init :-** Initialize a new repository in your project folder.
 - **git status:-** Check the status of files in the working directory (untracked, modified, etc.).
 - **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.
3. Work on a New Feature
 - **git branch master:-** Create a new branch for a feature (e.g., feature-1).
 - **git checkout master :-** Switch to the new branch.
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.
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.
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.
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.

```
admin@DESKTOP-AMCRBC2 MINGW64 ~/c24077 (main)
$ git pull origin main
From https://github.com/snehalparab27/demo
 * branch                main          -> FETCH_HEAD
Updating 1672d88..7df1fad
Fast-forward
 demo.txt | 2 ++
 1 file changed, 2 insertions(+)
```

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

 MINGW64:/c/Users/admin/c24077

```
admin@DESKTOP-AMCRBC2 MINGW64 ~/c24077 (master)
$ git init
Reinitialized existing Git repository in C:/Users/admin/c24077/.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>

```
C:\Users\admin>cd C:\Users\admin\c24077

C:\Users\admin\c24077>git clone https://github.com/snehalparab27/demo.git
Cloning into 'demo'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 10 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.

C:\Users\admin\c24077>_
```

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

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

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

`git branch <branch-name>`

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (main)
$ git branch new_branch
```

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>`

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (main)
$ git checkout new_branch
Switched to branch 'new_branch'

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (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 (*).

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)`

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

`git commit -m "Description of changes"`

`git`

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git add myfile.txt
fatal: pathspec 'myfile.txt' did not match any files

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git add myfile.txt
fatal: pathspec 'myfile.txt' did not match any files

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ ls
hello.txt  new.txt  test.txt

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git add hello.txt

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git add .

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git status
On branch branch1
nothing to commit, working tree clean

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git commit -m "decsription of changes"
On branch branch1
nothing to commit, working tree clean
```

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`

```

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (branch1)
$ git checkout master
branch 'master' set up to track 'origin/master'.
Switched to a new branch 'master'

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git pull origin
Already up to date.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git merge branch1
fatal: refusing to merge unrelated histories

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git init
Reinitialized existing Git repository in C:/Users/newgit/myrepo_sam79/.git/

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git merge branch1
fatal: refusing to merge unrelated histories

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git branch
  branch1
  main
* master
  new_branch

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git merge new_branch
fatal: refusing to merge unrelated histories

```

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

git pull origin main

3. **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.

```

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git merge branch1 --allow-unrelated-histories
Auto-merging hello.txt
CONFLICT (add/add): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master|MERGING)
$ git add hello.txt

```

Step 5: Resolving Merge Conflicts (If Any)

1. **Check for conflicts** If Git encounters conflicts during the merge, it will pause and mark the conflicted files.
2. **Open the conflicted files** Conflicted sections will be marked with:

```
<<<<<<< HEAD
```

```
(changes from `main` branch)
```

```
=====
```

```
(changes from `<branch-name>`)
```

```
>>>>>>> <branch-name>
```

3. **Resolve the conflicts** Edit the file to keep the changes you want and remove the conflict markers (<<<<<<<, =====, >>>>>>>).
4. **Mark the conflicts as resolved** After resolving conflicts, stage the files as resolved:

```
git add <resolved-file>
```

5. **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

1. **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.

Step 7: Clean Up (Optional)

1. **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
```

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

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

```

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master|MERGING)
$ git commit -m "Merged branch1 into master allowing unrelated histories"
[master f756b20] Merged branch1 into master allowing unrelated histories

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 401 bytes | 401.00 KiB/s, done.
Total 3 (delta 0), reused 1 (delta 0), pack-reused 0 (from 0)
To https://github.com/Shramikapatne20/myrepo_sam79.git
   8c490e8..f756b20  master -> master

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git branch -d branch1
Deleted branch branch1 (was 23cf05c).

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ ^[[200~git push origin --delete branch1
bash: $'\E[200~git': command not found

```

Step 8: Regular Maintenance

1. **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

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

Practical 3

AIM: Demonstrate Git for Collaboration

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
```

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
```

```

hp@DESKTOP-RL6G3J5 MINGW64 ~ (master)
$ cd "C:\Users\newgit"

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit (master)
$ git clone "https://github.com/Shramikapatne20/myrepo_sam79.git"
fatal: destination path 'myrepo_sam79' already exists and is not an empty direct
ory.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit (master)
$ cd myrepo_sam79

```

Step 3: Work on the Project Locally

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

1. **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`

```

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git checkout new_branch
Switched to branch 'new_branch'

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git add .

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git commit -m "Added some text"
[new_branch 749e265] Added some text
1 file changed, 1 insertion(+)

```

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

`git add . # Stages all modified files`

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

`git commit -m "Add feature X"`

Step 4: Push Changes to GitHub

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

- 1 **Push Your Changes** to the remote repository:

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

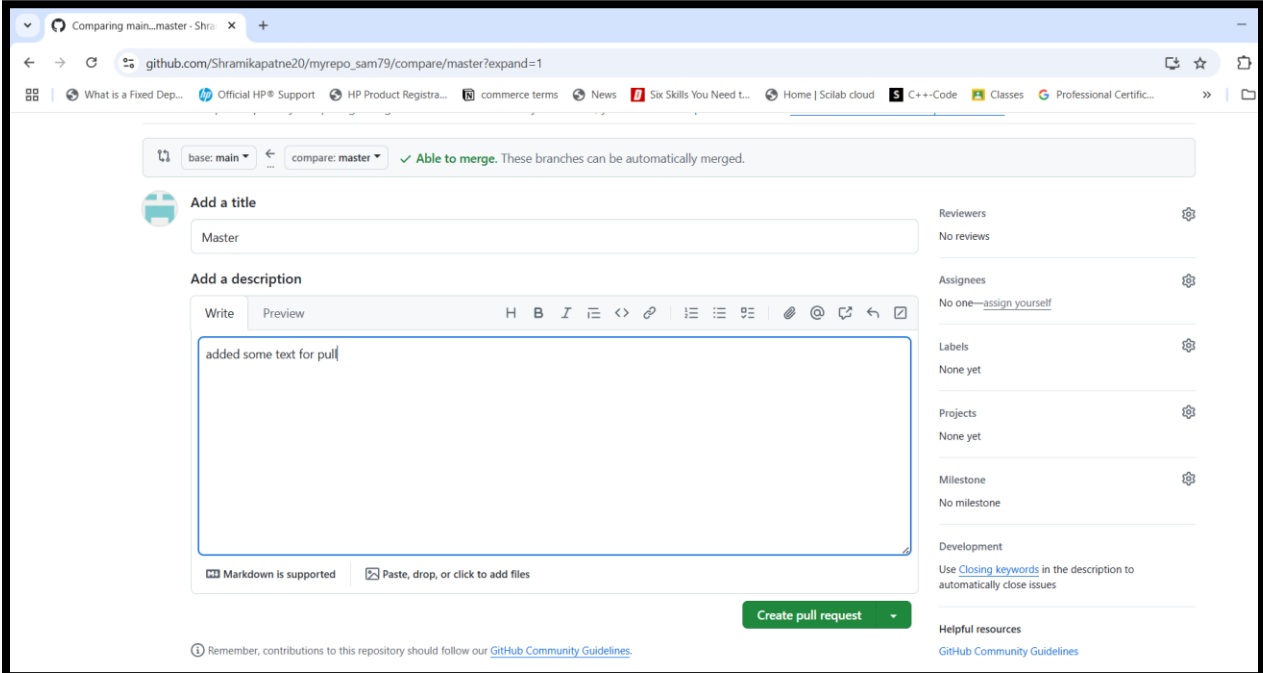
This uploads your local changes to your GitHub repository.

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git push origin master
Everything up-to-date
```

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).

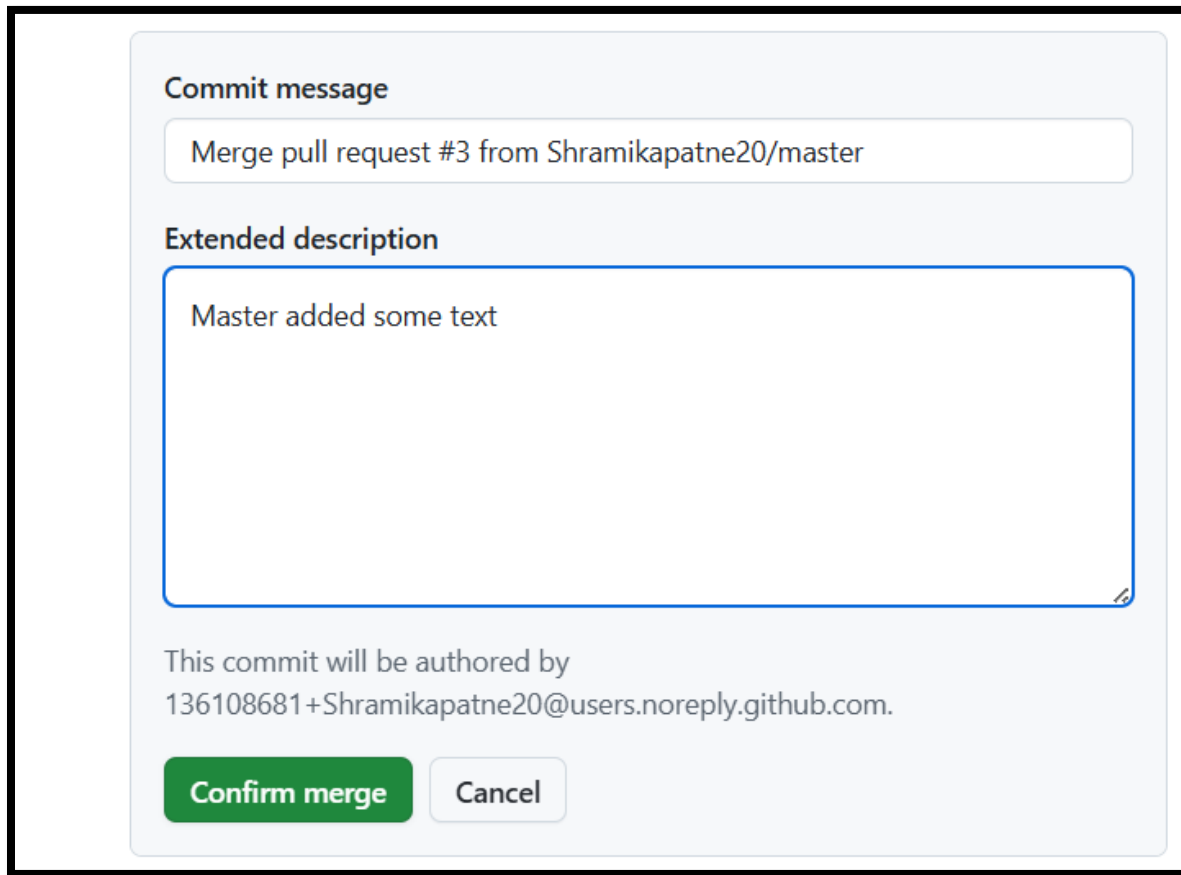
1. **Go to the GitHub Repository:** Visit the repository where you want to make the changes.
2. **Create a Pull Request:** On GitHub, you'll see an option to compare your branch (e.g., feature/your-feature) with the main branch of the repository (e.g., main). Click **New Pull Request**.
3. **Fill Out the Pull Request Form:**
 - Add a **title** and **description** explaining the changes you've made.
 - Review your changes.
 - Click **Create Pull Request**.



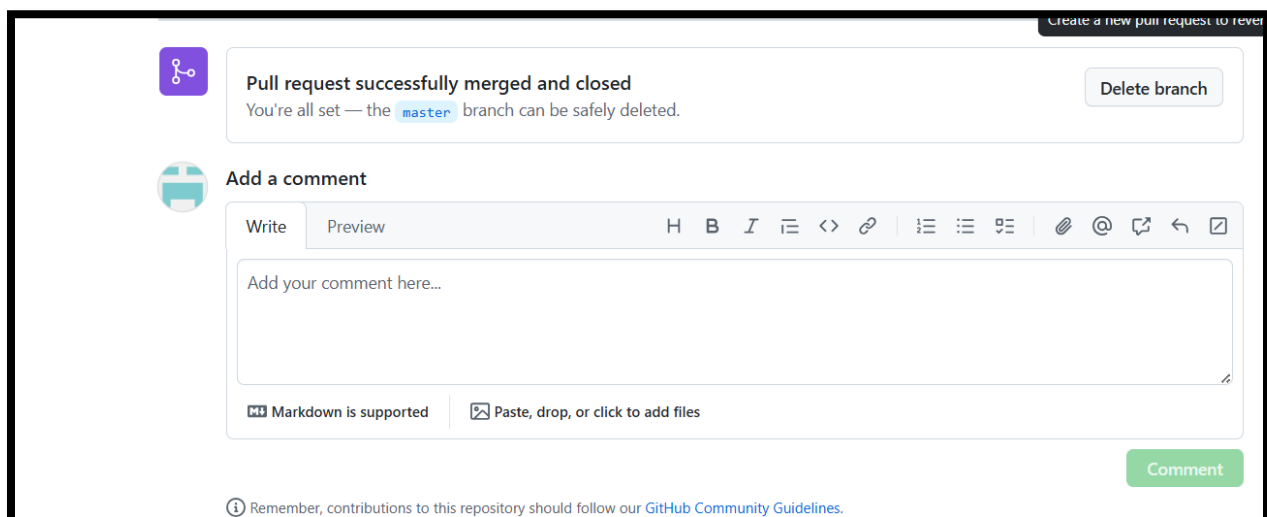
The screenshot shows the GitHub interface for creating a pull request. At the top, it indicates 'base: main' and 'compare: master' with a green checkmark stating 'Able to merge. These branches can be automatically merged.' Below this, there's a section for 'Add a title' with a text input field containing 'Master'. Underneath is 'Add a description' with a rich text editor containing 'added some text for pull'. To the right of the description field are several metadata sections: 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (Use Closing keywords in the description to automatically close issues). At the bottom right, there is a green 'Create pull request' button. A footer note mentions 'Remember, contributions to this repository should follow our GitHub Community Guidelines'.

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.



The screenshot shows the GitHub merge confirmation interface. At the top, there's a section titled "Commit message" with a text input field containing "Merge pull request #3 from Shramikapatne20/master". Below this is an "Extended description" section with a larger text area containing "Master added some text". At the bottom, it states "This commit will be authored by 136108681+Shramikapatne20@users.noreply.github.com." and features two buttons: "Confirm merge" (green) and "Cancel" (light blue).



The screenshot shows the GitHub pull request success message and comment section. At the top, a purple icon is next to the message "Pull request successfully merged and closed" and "You're all set — the master branch can be safely deleted." with a "Delete branch" button. Below this is the "Add a comment" section, which includes a "Write" tab, a "Preview" tab, and a rich text editor with various formatting options (bold, italic, link, etc.). The text area contains "Add your comment here...". At the bottom, there are links for "Markdown is supported" and "Paste, drop, or click to add files", and a green "Comment" button.

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
```

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
```

4. **Push the Changes to Your Fork:**

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

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
```

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.

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git remote add upstream https://github.com/snehalparab27/demo.git

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git fetch upstream
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 24 (delta 1), reused 5 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (24/24), 8.28 KiB | 184.00 KiB/s, done.
From https://github.com/snehalparab27/demo
 * [new branch]      branch1 -> upstream/branch1
 * [new branch]      main   -> upstream/main

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git push origin master
Everything up-to-date

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (main)
$ git pull origin master
From https://github.com/Shramikapatne20/myrepo_sam79
 * branch            master       -> FETCH_HEAD
Updating 23cf05c..f756b20
Fast-forward
 hello.txt | 2 +-
 myfile.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 myfile.txt
```

Practical 4

AIM: Demonstrate Collaborating and cloning using Git

Set Up Git and GitHub

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

Install Git: If you haven't already, download and install [Git](#) on your machine.

Create a GitHub Account: Go to [GitHub](#) and create an account if you don't have one.

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`

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.

4. **Find a Repository to Clone:** Visit the repository page on GitHub (e.g., <https://github.com/username/repository>) and click the green **Code** button.
5. **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`.
6. **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

```
hp@DESKTOP-RL6G3J5 MINGW64 ~ (master)
$ cd "C:\Users\newgit"

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit (master)
$ git clone "https://github.com/Shramikapatne20/myrepo_sam79.git"
fatal: destination path 'myrepo_sam79' already exists and is not an empty direct
ory.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit (master)
$ cd myrepo_sam79
```

Step 3: Work on the Project Locally

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

2. **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`

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git checkout new_branch
Switched to branch 'new_branch'

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git add .

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git commit -m "Added some text"
[new_branch 749e265] Added some text
1 file changed, 1 insertion(+)
```

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

`git add . # Stages all modified files`

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

`git commit -m "Add feature X"`

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`

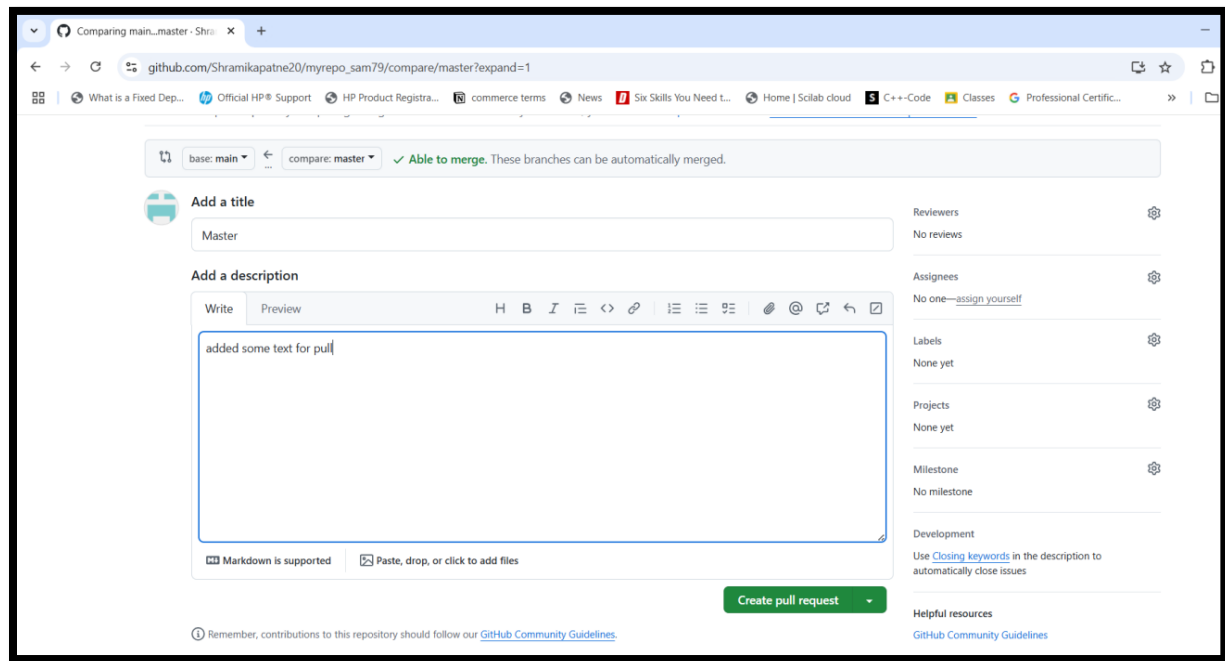
This uploads your local changes to your GitHub repository.

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git push origin master
Everything up-to-date
```

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).

6. **Go to the GitHub Repository:** Visit the repository where you want to make the changes.
7. **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**.
8. **Fill Out the Pull Request Form:**
 - Add a **title** and **description** explaining the changes you've made.
 - Review your changes.
 - Click **Create Pull Request**



9. **Code Review:** The repository maintainer (or other collaborators) will review your changes. They might ask for changes or approve the PR.
10. **Merge the Pull Request:** Once your changes are approved, the maintainer will merge your changes into the main branch of the project.

Commit message

Merge pull request #3 from Shramikapatne20/master

Extended description


Master added some text

This commit will be authored by
136108681+Shramikapatne20@users.noreply.github.com.

Confirm merge

Cancel


Create a new pull request to revert



Pull request successfully merged and closed

You're all set — the `master` branch can be safely deleted.











Delete branch



Add a comment

Write

Preview

H B I          

Add your comment here...

Markdown is supported

Paste, drop, or click to add files

Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

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:

5. **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.

6. **Fetch the Latest Changes from the Original Repository:**

```
git fetch upstream # Fetch the changes from the original repo
```

7. **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
```

8. **Push the Changes to Your Fork:**

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

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.

3. **Switch to Your Local Main Branch:**

```
git checkout main
```

4. **Pull Latest Changes from GitHub:**

```
git pull origin main # Pull the latest changes from the remote repository
```

Step 8: Collaborate with Other Developers

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

5. **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).
6. **Create Feature Branches:** Always create a new branch for each feature or bug fix. This avoids conflicts and keeps the history clean.
7. **Review Pull Requests:** If you're reviewing someone else's PR, ensure you provide feedback and approve it once you're satisfied.
8. **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.

```
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git remote add upstream https://github.com/snehalparab27/demo.git

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git fetch upstream
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 24 (delta 1), reused 5 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (24/24), 8.28 KiB | 184.00 KiB/s, done.
From https://github.com/snehalparab27/demo
* [new branch]      branch1 -> upstream/branch1
* [new branch]      main    -> upstream/main

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (new_branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git push origin master
Everything up-to-date

hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (master)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

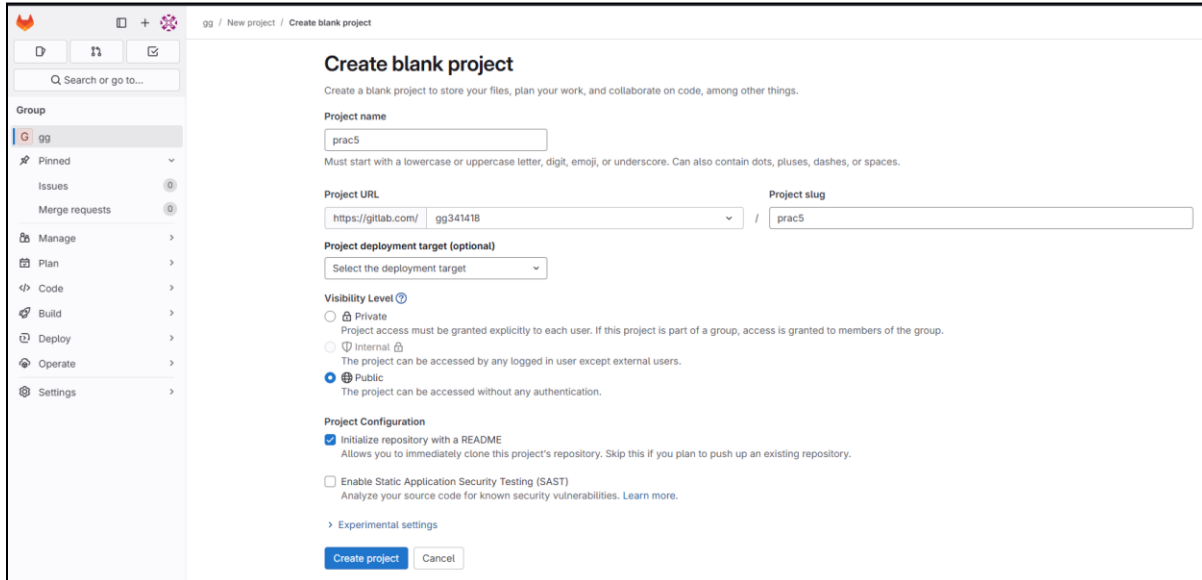
hp@DESKTOP-RL6G3J5 MINGW64 /c/Users/newgit/myrepo_sam79 (main)
$ git pull origin master
From https://github.com/Shramikapatne20/myrepo_sam79
* branch            master    -> FETCH_HEAD
Updating 23cf05c..f756b20
Fast-forward
 hello.txt | 2 +-
 myfile.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 myfile.txt
```

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.



The screenshot shows the 'Create blank project' page in GitLab. On the left is a sidebar with navigation options: Group, Pinned, Issues, Merge requests, Manage, Plan, Code, Build, Deploy, Operate, and Settings. The main area is titled 'Create blank project' and includes a description: 'Create a blank project to store your files, plan your work, and collaborate on code, among other things.' The form contains the following fields and options:

- Project name:** A text input field with 'prac5' entered.
- Project URL:** A dropdown menu showing 'https://gitlab.com/' and 'gg341418'.
- Project slug:** A text input field with 'prac5' entered.
- Project deployment target (optional):** A dropdown menu with 'Select the deployment target'.
- Visibility Level:** Three radio buttons: 'Private' (selected), 'Internal', and 'Public'.
- Project Configuration:** A section with two checkboxes: 'Initialize repository with a README' (checked) and 'Enable Static Application Security Testing (SAST)' (unchecked).
- Buttons:** 'Create project' and 'Cancel' at the bottom.

4. Create a file (index.html):

```
<html>
```


```
<body>
```

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

```
</body>
```

```
</html>
```

New file



The screenshot shows the 'New file' interface in the GitLab Web IDE. At the top, there are 'Cancel' and 'Commit changes' buttons. Below them is a file name input field with 'index.html' entered. The main area is a code editor showing the following HTML code:

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

5. Click Commit and push changes.

Commit changes

Commit message

Add new file

Branch

☒ Commit to the current `main` branch

☐ Commit to a new branch

Cancel

Commit changes

AnantKumbhar/dev

index.html - main - undefined

gitlab.com/gg341410/prac5/-/blob/main/index.html

Project

Search or go to...

Project

prac5

Pinned

Issues

Merge requests

Manage

Plan

Code

Merge requests

Repository

Branches

Commits

Tags

Repository graph

Compare revisions

Snippets

Build

Secure

Deploy

Operate

Monitor

Help

The file has been successfully created.

main / prac5 / index.html

index.html

Find file

Blame

Edit

Add new file

Anant Kumbhar authored right now

ac8ac825

History

index.html 76 B

1 <html>

2 <body>

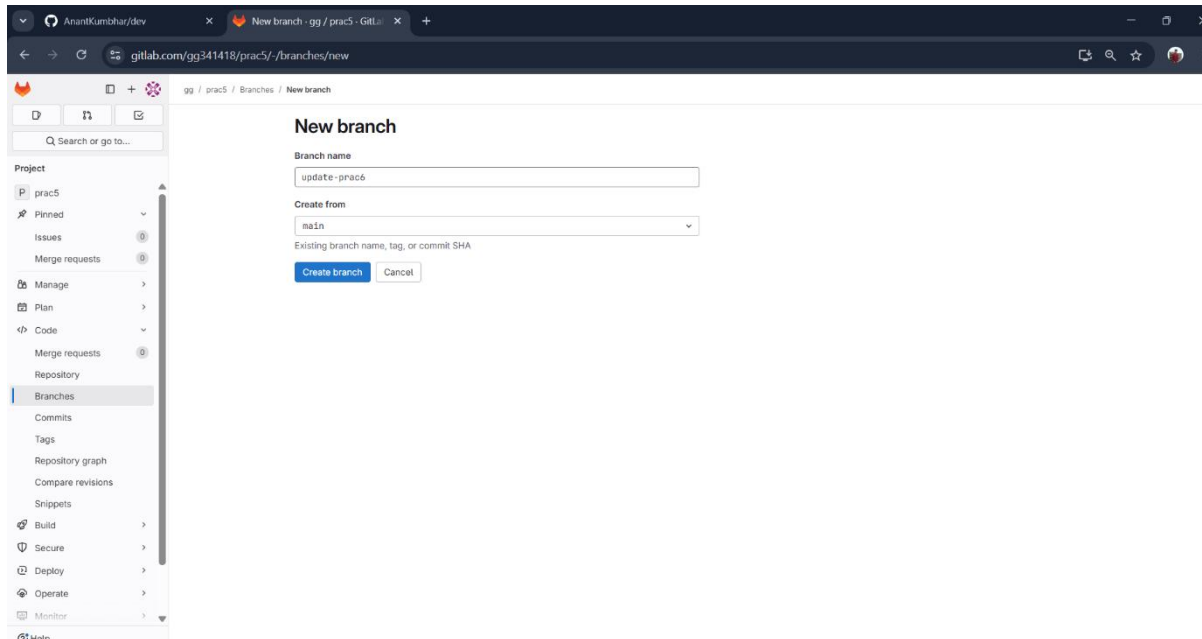
3 <div>Hello from GitLab</div>

4 </body>

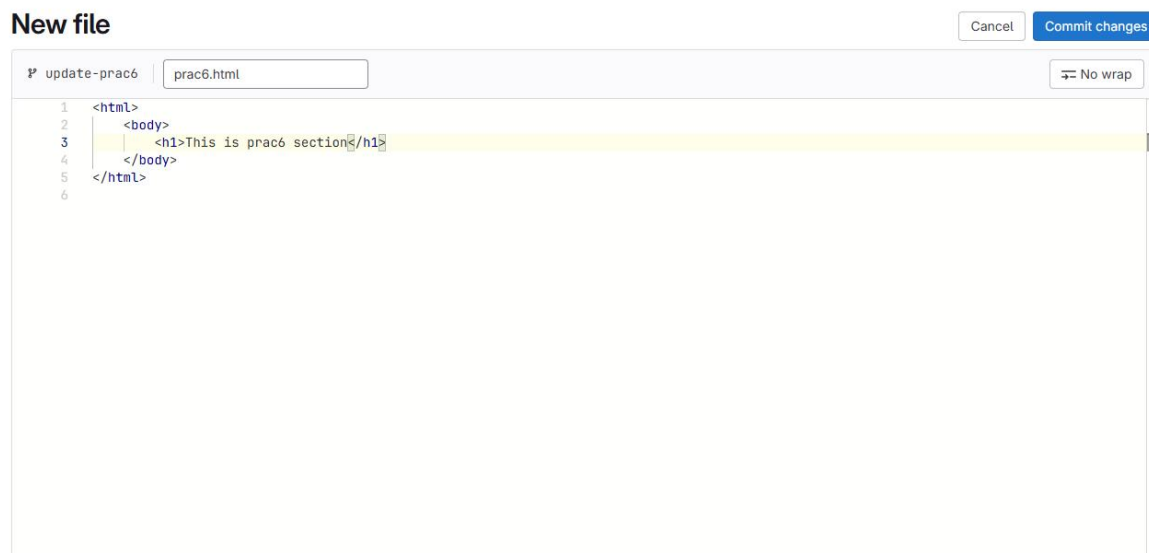
5 </html>

Performing merge requests using GitLab

1. Create a new branch in Web IDE.



2. Add /Edit a File and Commit



Commit changes

Commit message

Add new file

Branch

☒ Commit to the current `update-prac6` branch

☐ Commit to a new branch

Cancel

Commit changes

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

gg / prac5 / Merge requests / New merge request

New merge request

From `update-prac6` into `main` [Change branches](#)

Title (required)

Merge

☐ Mark as draft

Drafts cannot be merged until marked ready.

Description

Preview

Describe the goal of the changes and what reviewers should be aware of.

Switch to rich text editing

Add description templates to help your contributors to communicate effectively!

Assignee

Unassigned

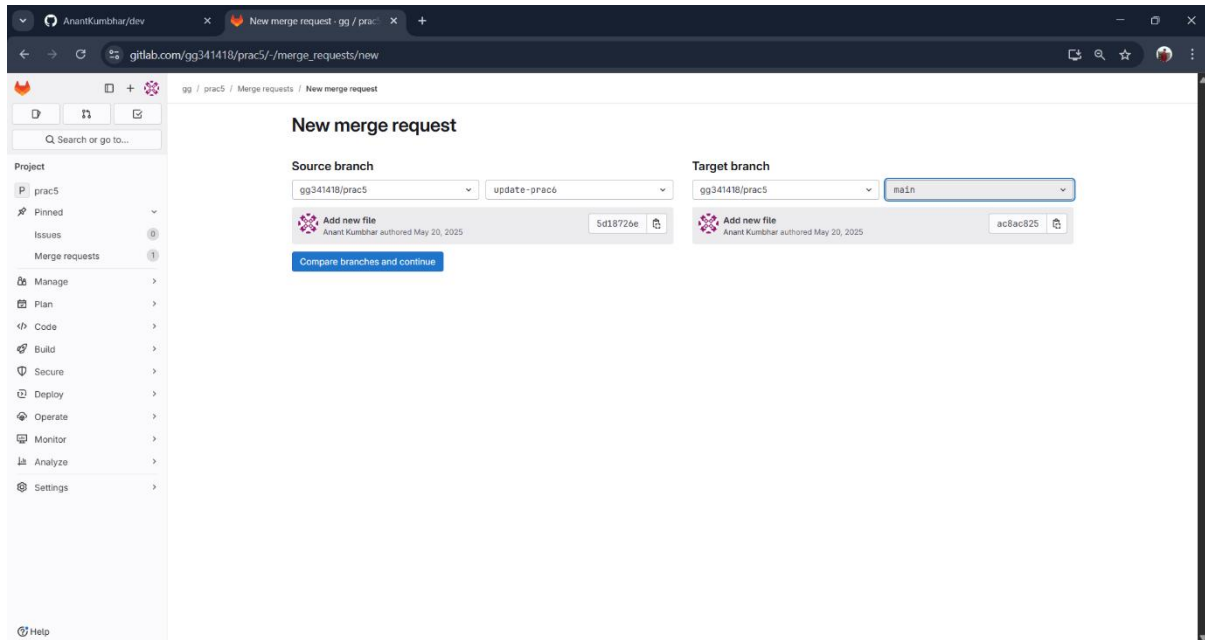
[Assign to me](#)

Reviewer



Unassigned

Approvals are optional.

Approval rules



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:

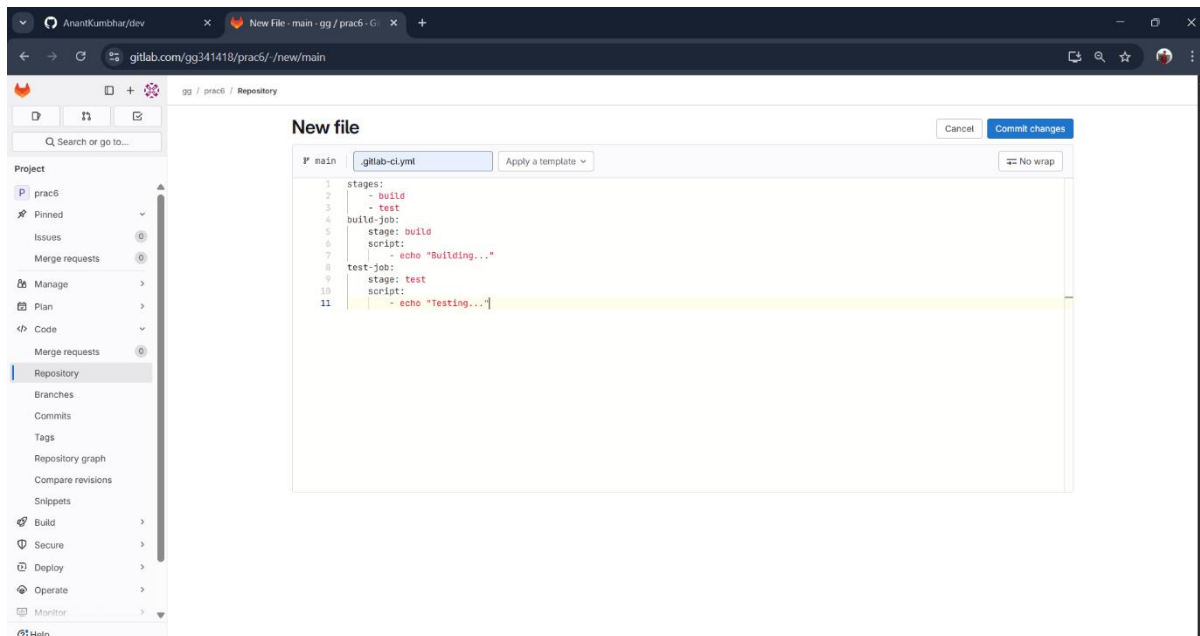
- echo "Building..."

test-job:

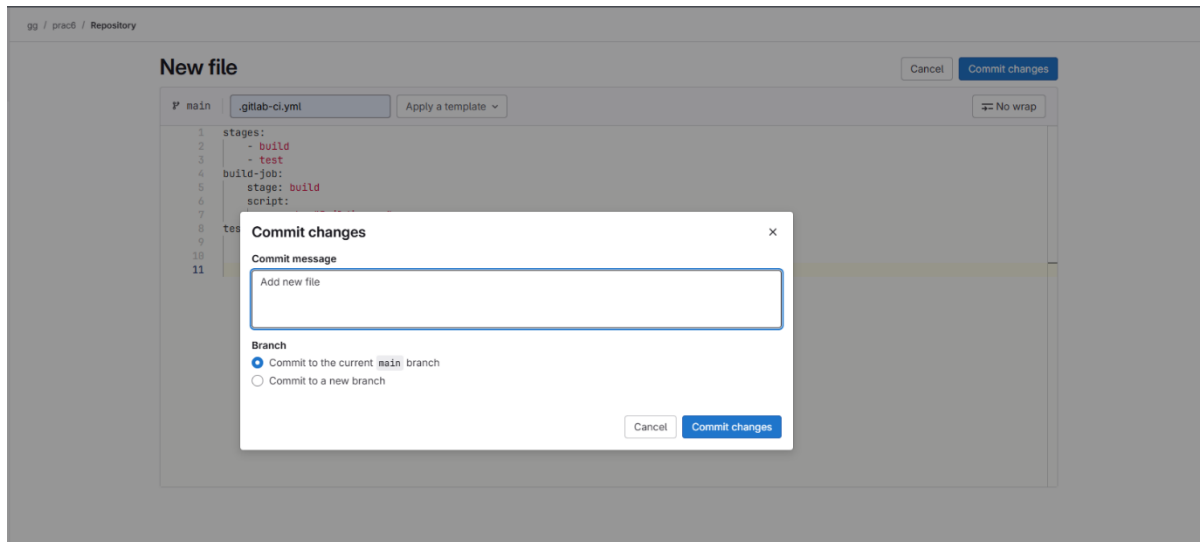
stage: test

script:

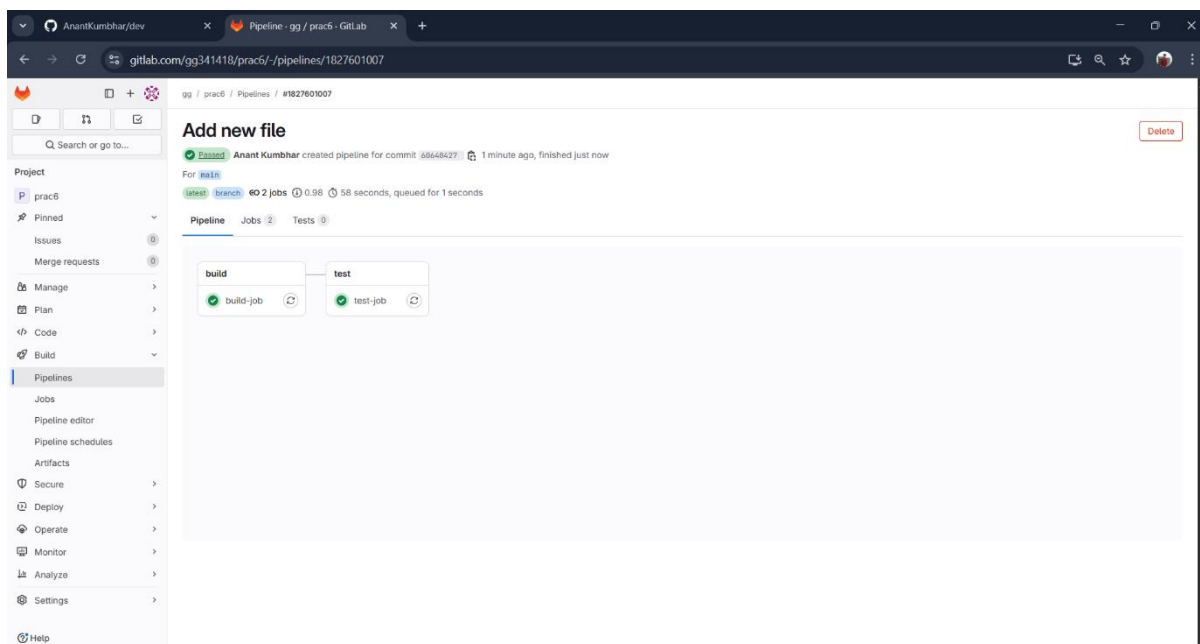
- echo "Testing..."



2. Commite and push



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



build-job

Passed Started 2 minutes ago by Anant Kumbhar

Search visible log output

```
1 Running with gitlab-runner 17.18.0-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 00:21
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@sha256:91627f55e8969086aab67d15c9
   2fb938508ff73948803da1330b8a853fecebb5 ...
7   ✓ Preparing environment 00:05
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 00:01
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 00:00
17  Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c9
   2fb938508ff73948803da1330b8a853fecebb5 ...
18  $ echo "Building..."
19  Building...
20  ✓ Cleaning up project directory and file based variables 00:01
21  Job succeeded
```

test-job

Passed Started 1 minute ago by Anant Kumbhar

Search visible log output

```
1 Running with gitlab-runner 17.18.0-pre.41.g5c23fd8e (5c23fd8e)
2   on blue-5.saas-linux-small-amd64.runners-manager.gitlab.com/default -AzERasQ, system ID: s_4cb09cee29e2
3   ✓ Preparing the "docker+machine" executor 00:20
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@sha256:91627f55e8969086aab67d15c9
   2fb938508ff73948803da1330b8a853fecebb5 ...
7   ✓ Preparing environment 00:04
8   Running on runner--azerasq-project-70015884-concurrent-0 via runner-azerasq-s-l-s-amd64-1747759901-d7593e3a...
9   ✓ Getting source from Git repository 00:01
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 00:01
17  Using docker image sha256:9981df1d883b246c27c62f8ccb9b57d3e07d14cee8092299e102b4a69c35ea61 for ruby:3.1 with digest ruby@sha256:91627f55e8969086aab67d15c9
   2fb938508ff73948803da1330b8a853fecebb5 ...
18  $ echo "Testing..."
19  Testing...
20  ✓ Cleaning up project directory and file based variables 00:00
21  Job succeeded
```


CI/Cd for python

Create script.py

```
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



 **Passed** Anant Kumbhar created pipeline for commit `b72008b4`  3 weeks ago, finished 3 weeks ago

For `main`

`branch`  1 job  0.48  28 seconds, queued for 2 seconds


Pipeline Jobs 1 Tests 0

test

 python_script 

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
```

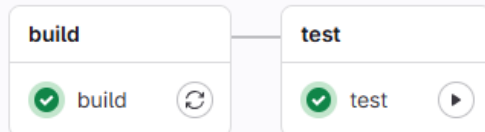
Commite and build pipeline

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

For main

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

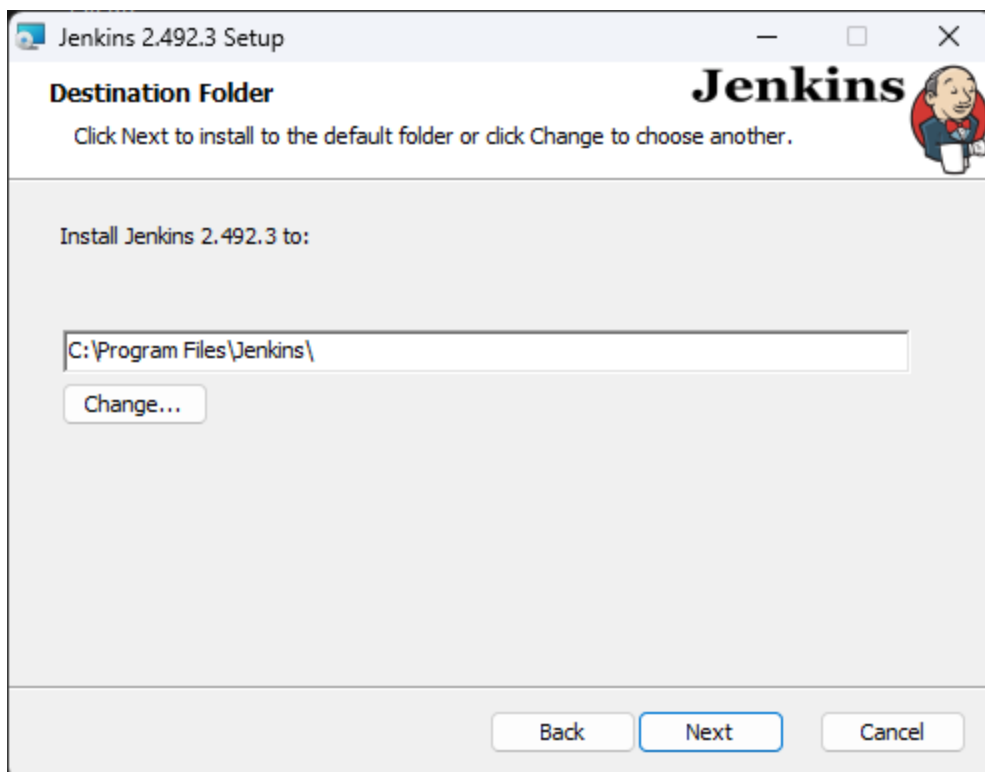
Pipeline Jobs 2 Tests 0



```
812 update alternatives. using /usr/lib/jvm/java-17-openjdk-amd64/bin/jconsole
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 pipline.



Jenkins 2.492.3 Setup

Jenkins

Port Selection

Choose a port for the service.

Please choose a port.

Port Number (1-65535):

8085

Test Port

⚠ Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.

BackNextCancel

Jenkins 2.492.3 Setup

Jenkins

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.492.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.492.3 to run successfully.

Logon Type:

☒ Run service as LocalSystem (not recommended)

☐ Run service as local or domain user:

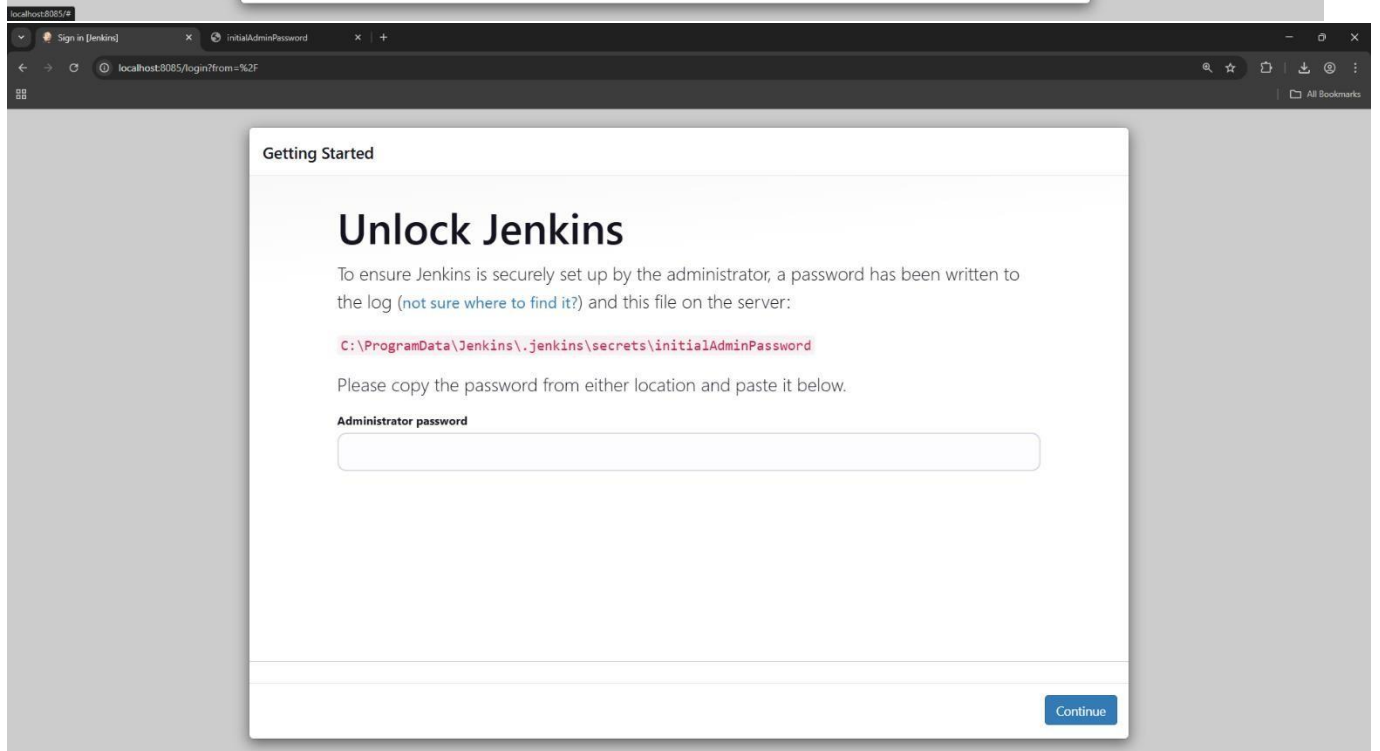
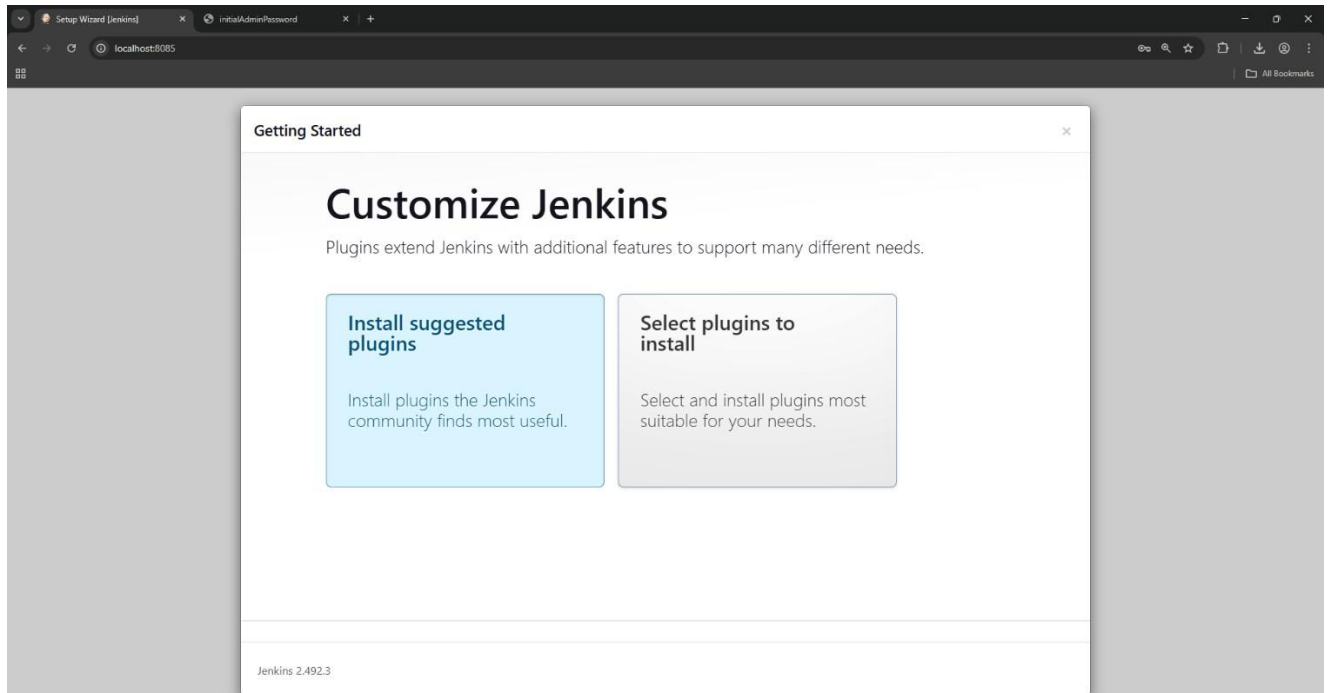
Account:

Password:

Test Credentials

BackNextCancel





Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle
🔍 Pipeline	🔍 GitHub Branch Source	🔍 Pipeline: GitHub Groovy Libraries	🔍 Pipeline Graph View
🔄 Git	🔄 SSH Build Agents	🔍 Matrix Authorization Strategy	🔍 PAM Authentication
🔄 LDAP	🔍 Email Extension	✓ Mailer	🔍 Dark Theme

** 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
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
Mailer
 ** Pipeline: Basic Steps
Gradle
 ** Pipeline: Milestone Step
 ** - required dependency

Instance Configuration

Jenkins URL:

http://localhost:8085/

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 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


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.
- 

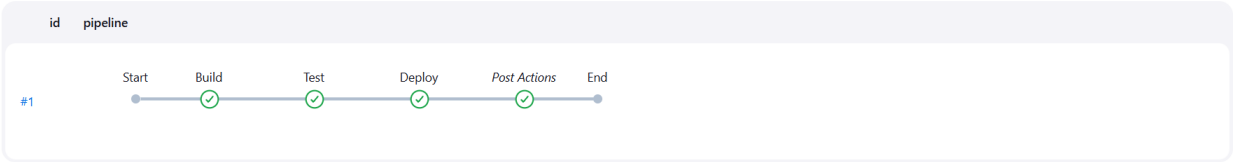
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
- 

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Build Demopipeline

[▶ Build](#) [Configure](#)



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);
```

```
}
```

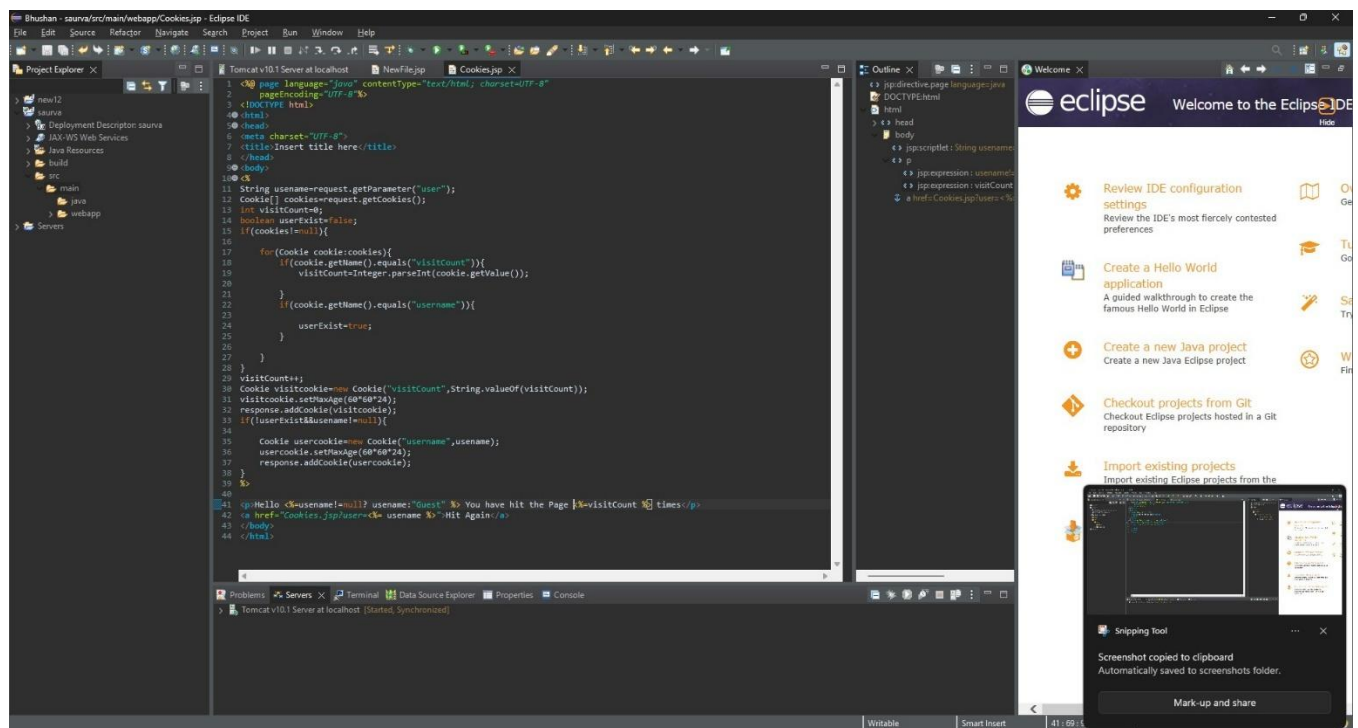
```
%>
```

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

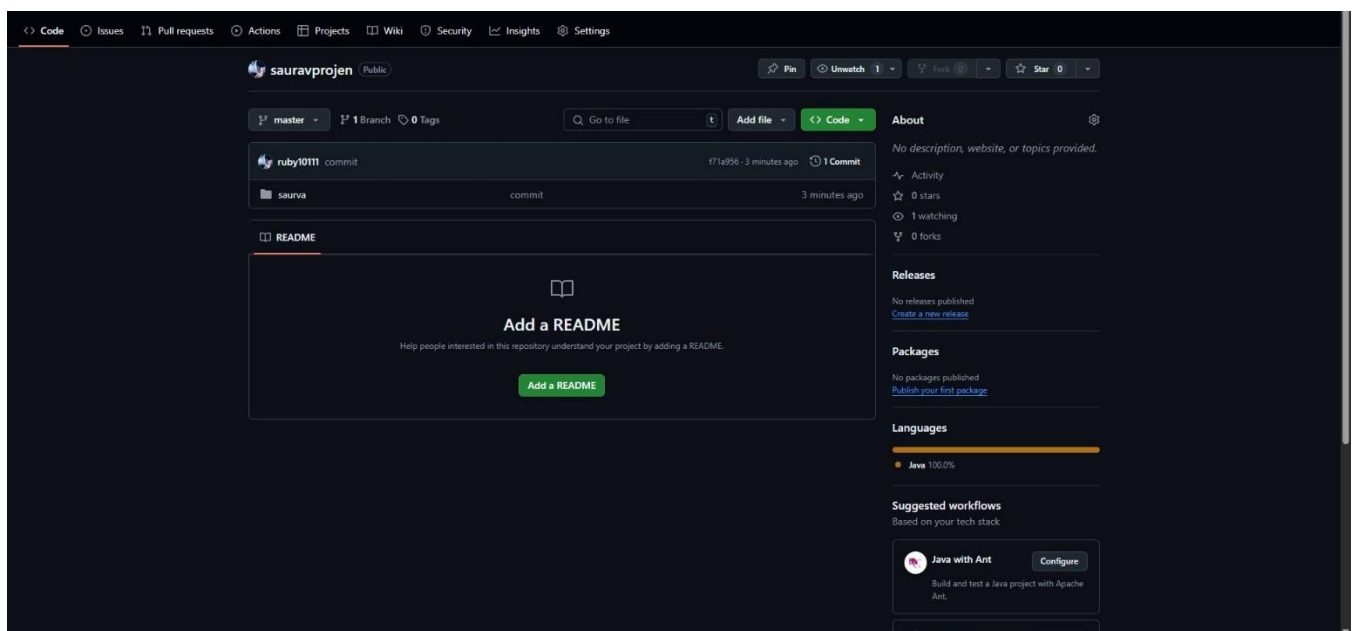
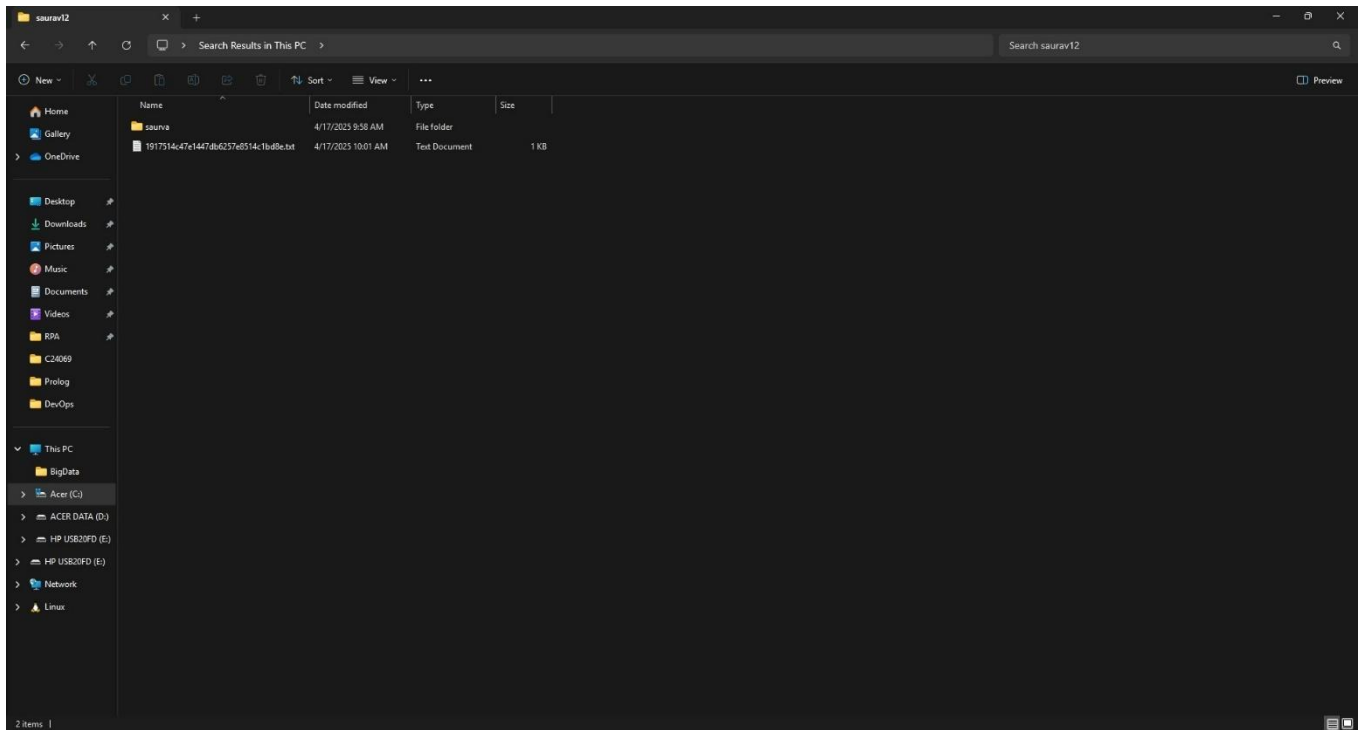
```
<a href="Cookies.jsp?user=<%= username %>">Hit Again</a>
```

```
</body>
```

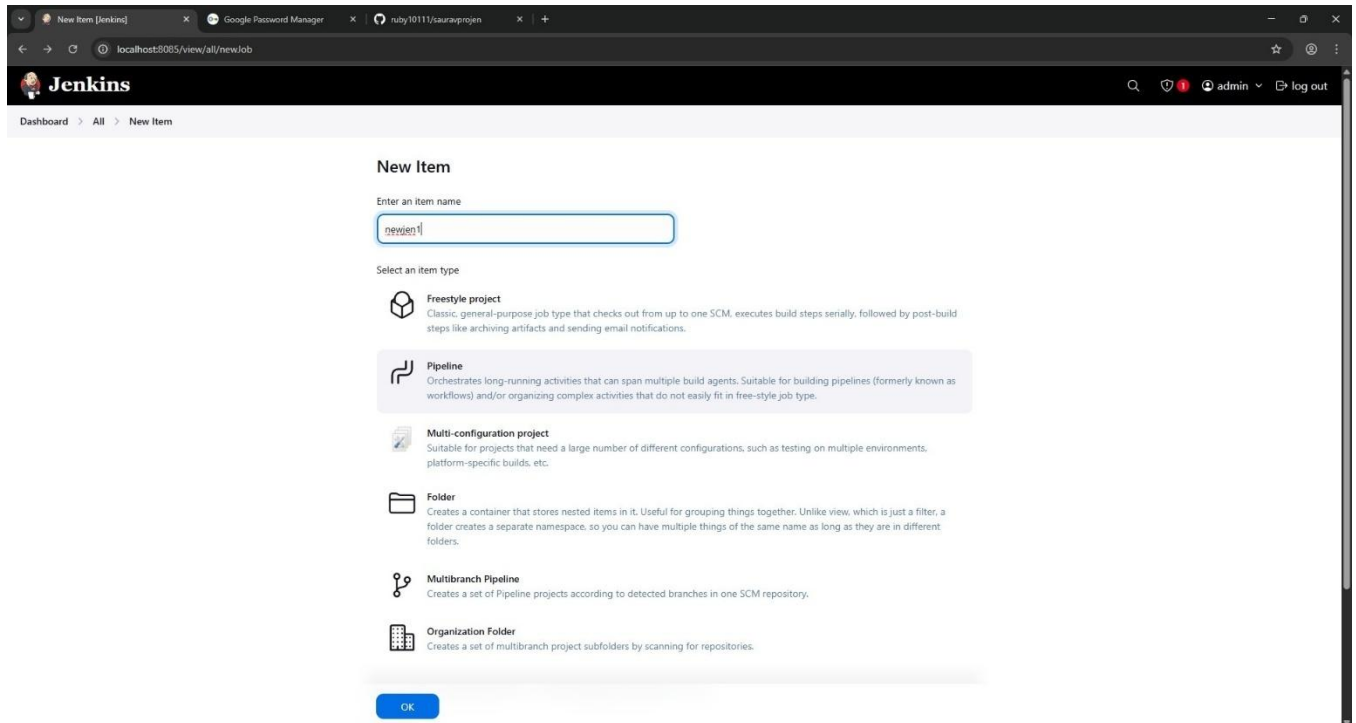
```
</html>
```



Push project on github



Create a pipeline:



```
pipeline{ agent 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"  
}  
}  
}
```

newjen1 Config [Jenkins] x Google Password Manager x ruby1011/sauravprojen x +

localhost:8085/job/newjen1/configure

Dashboard > newjen1 > Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

☐ Enter next trigger for this pipeline

☐ Poll SCM

☐ Trigger builds remotely (e.g., from scripts)

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

```
script{
  19 def srcPath="saurva/src/main/webapp"
  20
  21 def destPath="C:\\Program Files\\Apache Software Foundation\\Tomcat 10.1\\webapps\\NewFile"
  22
  23 if(fileExists(srcPath)){
  24     bat"xcopy /E /I \\${srcPath} \\${destPath}"
  25
  26 }
  27 else{
  28     error "Source directory ${srcPath} does not exists"
  29
  30 }
  31 }
  32
  33 }
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Save Apply

newjen1 #9 [Jenkins] x Google Password Manager x ruby1011/sauravprojen x +

localhost:8085/job/newjen1/9/

Jenkins

Dashboard > newjen1 > #9

Status

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

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

Add description Keep this build forever

Started by user admin

This run spent:

- 7 ms waiting;
- 2.6 sec build duration;
- 2.6 sec total from scheduled to completion.

Revision: f71a956b127acf4ee9bdf09333775eab88b0ec42

Repository: <https://github.com/ruby1011/sauravprojen>

- refs/remotes/origin/master

No changes.

Started 1 min 12 sec ago
Took 2.6 sec

Dashboard >

+ New Item

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

Add description

All +

S	W	Name	Last Success	Last Failure	Last Duration
		Demopipeline	7 days 22 hr #2	N/A	1.6 sec

Icons: S W L

...

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/
lsstat /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

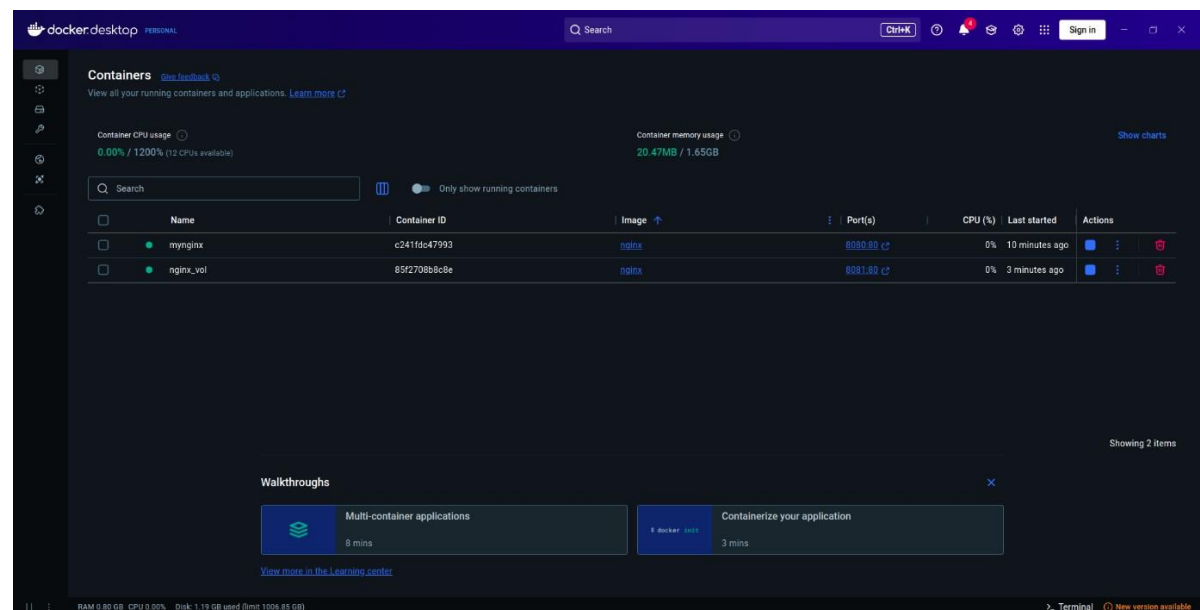
Now any data added to the /usr/share/nginx/html inside the container will persist even if the container is removed.

```
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:eb7c28f99ff6e48b821ddd884433bb48c5e0cafbcc33be2444270361ebdaa3c
=> => 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`

```
ubuntu@ubuntu:~/DevOps$ 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`

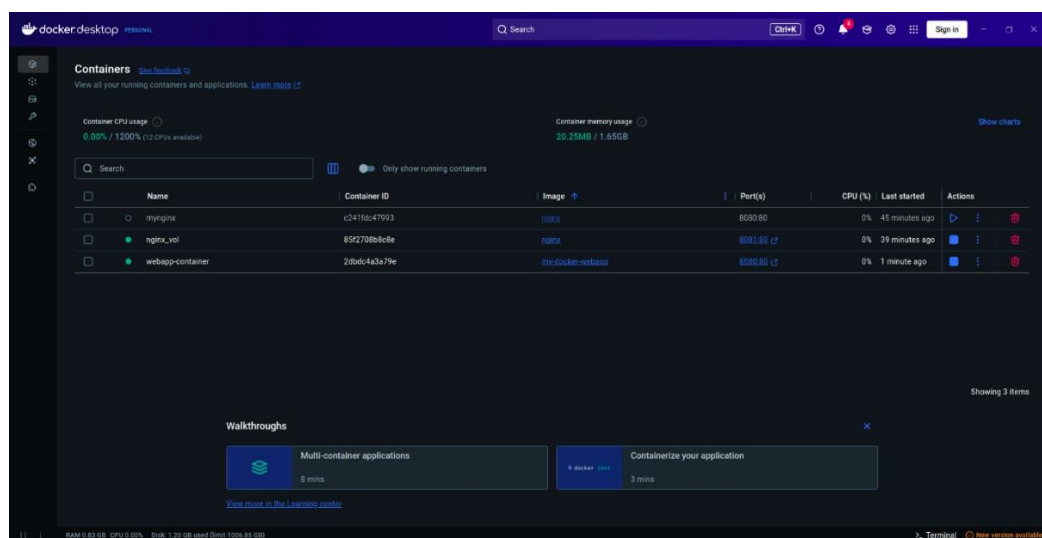
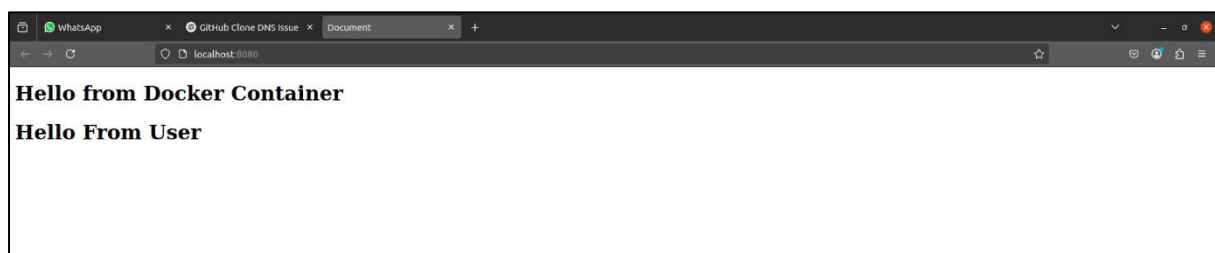
```
ubuntu@ubuntu:~/DevOps$ docker stop webapp-container
webapp-container
```

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:eb7c28f99ff6e48b821ddd884433bb48c5e0cafbfcc33be2444270361ebdaa3c
```



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:6 http://ppa.launchpad.net/rock-core/qt4/ubuntu focal InRelease
Hit:7 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Hit:8 http://ln.archive.ubuntu.com/ubuntu focal InRelease
Hit:9 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Hit:10 http://ln.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:11 http://ln.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-l386/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
  libwireshark13 libwsutil11 libxext-dev qt5-qmake qt5-qmake-bin qtbase5-dev qtbase5-dev-tools x11proto-next-dev
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
  libavformat58 python2.7-minimal vlc-plugin-samba python3-librawscale3 vlc-plugin-qt libzmq5
  python2.7-minimal vlc-plugin-skins2 vlc-plugin-visualization vlc-l10n
  libcgmath6 libpython2.7 python2.7 vlc-plugin-notify libvlc5 python3-ipython
  libpython2.7-dev libgvc6-plugins-gtk libpostproc55 liblab-gamut1 liblccore9
  libvlc-bin libmpx2 libzvbi0 libavcodec58 vlc libcdt5 libavutil56 vlc-data
  libpathplan4 libavdevice58 librawscale5 libgpr2 libstdc2-2.0-0 libmysofa1
  inetutils-traceroute vlc-plugin-video-splitter libpython2.7-minimal
  libgraphviz-dev libgvc6 vlc-plugin-base libpython2.7-stdlib traceroute
  libzvt-common graphviz
Learn more about Ubuntu Pro at https://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-and64-bin grub-efi-and64-signed libarchive13 libcryptsetup12 libjuh-java libjurt-java
  libmysqldbent21 libpoppler-cppv5 libpoppler-glib8 libpoppler97 libraw19 libreoffice-base-core libreoffice-calc libreoffice-common libreoffice-core libreoffice-draw libreoffice-gnome
16 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://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jinja2 all 2.10.1-2ubuntu0.6
Get:2 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 python3-crypto amd64 2.6.1-13ubuntu2 [237 kB]
Get:3 http://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-dnspython all 1.16.0-1ubuntu1 [89.2 kB]
Get:4 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 ieee-data all 20180805.1 [1,589 kB]
Get:5 http://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-netaddr all 0.7.19-2ubuntu1 [236 kB]
Get:6 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 ansible all 2.9.6dfsg-1 [5,794 kB]
Get:7 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-argcomplete all 1.8.1-1.3ubuntu1 [27.2 kB]
Get:8 http://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jmespath all 0.9.4-2ubuntu1 [21.5 kB]
Get:9 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-kerberos amd64 1.1.14-3.1build1 [22.6 kB]
Get:10 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-libcloud all 2.8.0-1 [1,403 kB]
Get:11 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-ntlm-auth all 1.1.0-1 [19.6 kB]
Get:12 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:13 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-ntlm all 1.1.0-1 [6,064 B]
Get:14 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-selinux amd64 3.0-1build2 [139 kB]
Get:15 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-xmldict all 0.12.0-1 [12.6 kB]
Get:16 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-wlrm all 0.3.0-2 [21.7 kB]
Get:1 http://ln.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 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
  libwireshark13 libwsutil11 libxext-dev qt5-qmake qt5-qmake-bin qtbase5-dev qtbase5-dev-tools x11proto-next-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-crypto python3-dnspython python3-jinja2 python3-jmespath python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
  python3-requests-kerberos python3-requests-ntlm python3-selinux python3-wlrm python3-xmldict
Suggested packages:
  cowsay sshpass python-jinja2-doc python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python3-argcomplete python3-crypto python3-dnspython python3-jinja2 python3-jmespath python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
  python3-requests-kerberos python3-requests-ntlm python3-selinux python3-wlrm 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://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jinja2 all 2.10.1-2ubuntu0.6
Get:2 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 python3-crypto amd64 2.6.1-13ubuntu2 [237 kB]
Get:3 http://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-dnspython all 1.16.0-1ubuntu1 [89.2 kB]
Get:4 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 ieee-data all 20180805.1 [1,589 kB]
Get:5 http://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-netaddr all 0.7.19-2ubuntu1 [236 kB]
Get:6 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 ansible all 2.9.6dfsg-1 [5,794 kB]
Get:7 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-argcomplete all 1.8.1-1.3ubuntu1 [27.2 kB]
Get:8 http://ln.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jmespath all 0.9.4-2ubuntu1 [21.5 kB]
Get:9 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-kerberos amd64 1.1.14-3.1build1 [22.6 kB]
Get:10 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-libcloud all 2.8.0-1 [1,403 kB]
Get:11 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-ntlm-auth all 1.1.0-1 [19.6 kB]
Get:12 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:13 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-ntlm all 1.1.0-1 [6,064 B]
Get:14 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-selinux amd64 3.0-1build2 [139 kB]
Get:15 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-xmldict all 0.12.0-1 [12.6 kB]
Get:16 http://ln.archive.ubuntu.com/ubuntu focal/universe amd64 python3-wlrm all 0.3.0-2 [21.7 kB]
Get:1 http://ln.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"
}
```

2. Check uptime

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

```

root@ubuntu:~# distribution local -t host,lnl -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 2:40, 1 user, load average: 1.08, 0.98, 0.90
root@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 --apt -a "name=nginx state=present update_cache=yes" --become\n[DEPRECATION WARNING]: Distribution Ubuntu 18-04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future 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.\nlocalhost | CHANGED =>\n    \"ansible_facts\": {\n        \"discovered_interpreter_python\": \"/usr/bin/python\"\n    },\n    \"cache_update_time\": 1747566323,\n    \"cache_updated\": true,\n    \"changed\": true,\n    \"stderr\": \"\",\n    \"stdout_lines\": [],\n    \"stdout\": \"Reading package lists...\\nBuilding dependency tree...\\nReading state information...\\nThe following packages were automatically installed and are no longer required:\\n chromium-m codecs-fmpg-extstreamer-gio vaapi\\n libgstbase-plugins-bad1.0-0 libgnutls-concurrent libidnopenhg5-dev libatsc2t5-sqlite libdvdtests libvulkan-dev libwifreshark3 libwiretap1 libxftlib libxext-dev qt5-qmake qt5-qnc-bin atbses-dev tools xiproot-xslt-dev\\n[note] autotemove\" to remove them.[info] The following additional dependencies will be installed-mod-http-inage-filter libnghttp-ng mod-http-xslt-filter\\n libnghttp-ng-mail libnghttp-ng-stream nginx-common nginx-core\\nsuggested packages:\\n fcgiwrap nghttp-dcl\\nthe following NEW packages will be installed:\\n libnghttp-ng http-inage-filter libnghttp-ng-mod-http-xslt-filter\\n libnghttp-ng-mod-mail libnghttp-ng-mod-stream nginx-common nginx-core\\no upgraded , 7 newly installed, 0 to remove and 67 not upgraded./Need to get 695 kB of archives./After this operation, 2141 kB of additional disc space will be used./Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 nginx-common all 1.18.0-0ubuntu1.7 [37.8 kB]/Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 libnghttp-ng-mod-http-inage-filter amd64 1.18.0-0ubuntu1.7 [14.8 kB]/Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 libnghttp-ng-mod-http-xslt-filter amd64 1.18.0-0ubuntu1.7 [13.1 kB]/Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 libnghttp-ng-mod-mail amd64 1.18.0-0ubuntu1.7 [43.0 kB]/Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 libnghttp-ng-mod-stream amd64 1.18.0-0bu bu nt1.7 [7.3 kB]/Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 nginx-core amd64 1.18.0-0ubuntu1.7 [425 kB]/Get::it! http://in.archive.ubuntu.c om/ubuntu focal-updates/main amd64 nginx all 1.18.0-0ubuntu1.7 [3620 B]\\nPreconfiguring packages ... ./fetched 65 kB in (85.6 kbps)/selecting previously unselected package nginx-common./\\n(Reading database ... \\n(Reading database ... 45%)/(Reading database ... 45%)/(Reading database ... 15%)/(Reading database ... 28%)/(Reading database ... 30%)/(Reading database ... 35%)/(Re ading database ... 40%)/(Reading database ... 40%)/(Reading database ... 50%)/(Reading database ... 55%)/(Reading database ... 62%)/(Reading database ... 65%)/(Reading database ... 70%)/(Re ading database ... 75%)/(Reading database ... 80%)/(Reading database ... 85%)/(Reading database ... 90%)/(Reading database ... 95%)/(Reading database ... 100%)/(Reading database ... 222461 fl es and directories currently installed.)/(Preparing to unpack .../0-nghttp-common_1.18.0-0ubuntu1.7_all.deb ...) /(Unpacking nginx-common (1.18.0-0ubuntu1.7) ...) /(Selecting previously un selected package libnghttp-ng-mod-http-inage-filter.(Preparing to unpack .../1-libnghttp-ng-mod-http-inage-filter_1.18.0-0ubuntu1.7_and64.deb ...) /(Unpackaging libnghttp-ng-mod-http-filter (1.18 .0-0ubuntu1.7)) /(Removing previously unselected package libnghttp-ng-mod-http-xslt-filter.(Preparing to unpack .../2-libnghttp-ng-mod-http-xslt-filter_1.18.0-0ubuntu1.7_and64.deb ...) /(Unpa cking libnghttp-ng-mod-http-xslt-filter (1.18.0-0ubuntu1.7) ...) /(Selecting previously unselected package libnghttp-ng-mod-mail.(Preparing to unpack .../3-libnghttp-ng-mod-mail_1.18.0-0ubuntu1.7 _and64.deb ...) /(Unpackaging libnghttp-ng-mod-mail (1.18.0-0ubuntu1.7) ...) /(Selecting previously unselected package libnghttp-ng-mod-stream.(Preparing to unpack .../4-libnghttp-ng-mod-strea m_1.18.0-0ub u nt1.7_and64.deb ...) /(Unpackaging libnghttp-ng-mod-stream (1.18.0-0ubuntu1.7) ...) /(Selecting previously unselected package nginx-core).(Preparing to unpack .../5/nginx-core_1.18.0-0ubu nt1.7_and64.deb ...) /(Ununpacking nginx-core (1.18.0-0ubuntu1.7) ...) /(Preparing to unpack .../6-nginx_1.18.0-0ubuntu1.7_all.deb ...) /(Nnpacking n ginx (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.s ystem.d/nginx.conf)./(Setting up libnghttp-ng-mod-http-filter (1.18.0-0ubuntu1.7) ...) /(Setting up libnghttp-ng-mod-http-xslt-filter (1.18.0-0ubu nt1.7) ...) /(Setting up libnghttp-ng-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 s ystem (245.4 ubuntu24) ...) /(Processing triggers for man-db (2.9.1-1) ...) /(Processing triggers for uwf (0.36-0ubuntu1)) ...) /(r/n\",
```


PRACTICAL-12

Aim: Demonstrate Ansible Playbooks

Install and Start Nginx

install_nginx.yml:

- name: Install and start Nginx on web servers hosts: webserver

become: true tasks:

- name: Install Nginx apt:

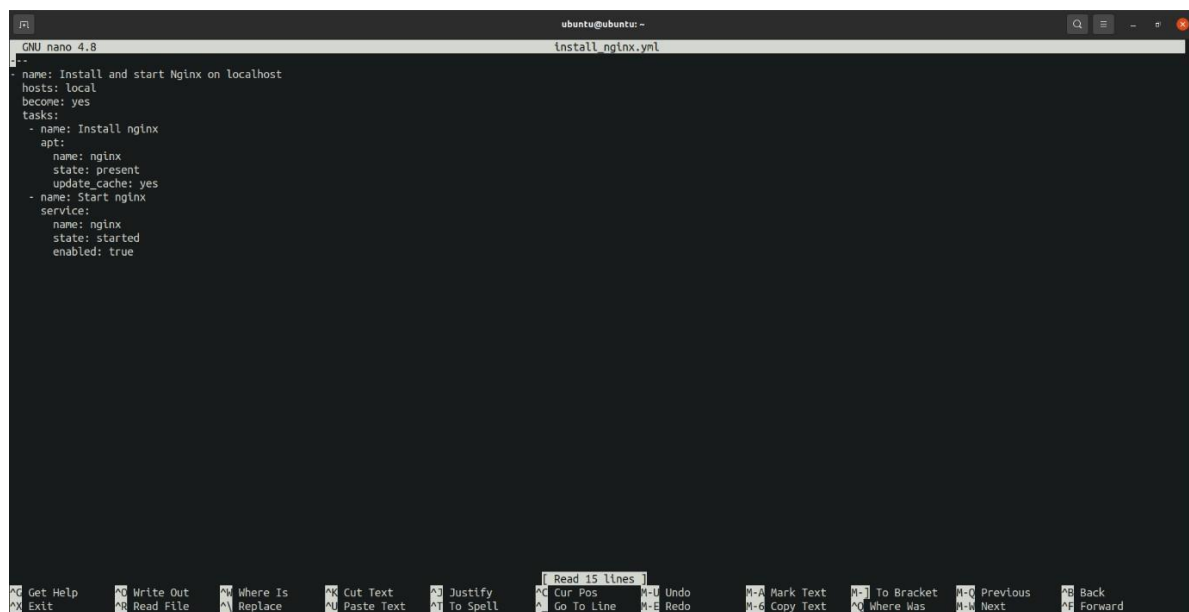
name: nginx state: present

update_cache: yes

- name: Start Nginx service:

name: nginx state: started enabled: true

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



```
GNU nano 4.8                                install_nginx.yml
-
- 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
```

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
```

