

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 NO.1

Basic Git commands

1. Check git
version git
-version

```
ubuntu@ubuntu:~$ git --version
git version 2.25.1
ubuntu@ubuntu:~$
```

2. Create folder and initilaze.

```
ubuntu@ubuntu:~$ git --version
git version 2.25.1
ubuntu@ubuntu:~$ mkdir newuser
ubuntu@ubuntu:~$ cd newuser/
ubuntu@ubuntu:~/newuser$ git init
Initialized empty Git repository in /home/ubuntu/newuser/.git/
ubuntu@ubuntu:~/newuser$
```

3. Configure Git
git config --global user.name "usernewncrd"
git config --global user.email "symca669@gmail.com"

```
ubuntu@ubuntu:~/newuser$ git config --global user.name "usernewncrd"
ubuntu@ubuntu:~/newuser$ git config --global user.email "symca669@gmail.com"
ubuntu@ubuntu:~/newuser$
```

4. Create a new project folder
mkdir git-demo
cd git-demo

```
ubuntu@ubuntu:~/newuser$ mkdir git-demo
ubuntu@ubuntu:~/newuser$ cd git-demo/
ubuntu@ubuntu:~/newuser/git-demo$
```

5. git init

```
ubuntu@ubuntu:~/newuser/git-demo$ git init
Initialized empty Git repository in /home/ubuntu/newuser/git-demo/.git/
ubuntu@ubuntu:~/newuser/git-demo$
```

6. Create and track a file:

```
echo "Hello User" >
file.txt git add file.txt
git commit -m "Initial commit"
```

```
ubuntu@ubuntu:~/newuser/git-demo$ echo "Hello User"> file.txt
ubuntu@ubuntu:~/newuser/git-demo$ git add file.txt
ubuntu@ubuntu:~/newuser/git-demo$ git commit -m "Initial Commit"
[master (root-commit) 5da5867] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 file.txt
ubuntu@ubuntu:~/newuser/git-demo$
```

7. Check status and

```
log: git status
git log
```

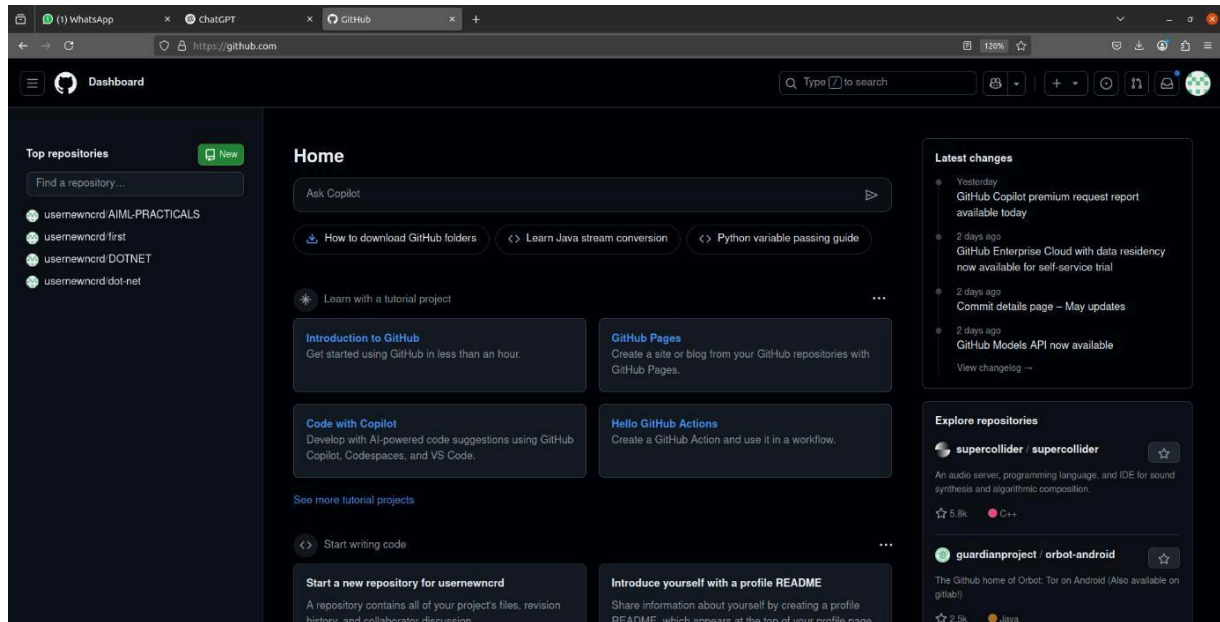
```
ubuntu@ubuntu:~/newuser/git-demo$ git status
On branch master
nothing to commit, working tree clean
ubuntu@ubuntu:~/newuser/git-demo$ git log
commit 5da586754b11433e7ab5ed5d1eafad9ad22d9289 (HEAD -> master)
Author: usernewncrd <symca669@gmail.com>
Date: Sun May 18 13:52:53 2025 +0530

    Initial Commit
ubuntu@ubuntu:~/newuser/git-demo$
```

PRACTICAL NO.2

Create and fork repositories in GitHub. Apply branch, merge, rebase concepts.

1. Create a GitHub account and log in.



2. Create a repository on GitHub (e.g., git-practice).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository](#)

Required fields are marked with an asterisk (*).

Owner *

Repository name *

usernewncrd

/ git-practice

git-practice is available.

Great repository names are short and memorable. Need inspiration? How about **studious-octo-palm-tree** ?

Description (optional)

☒ Public

Anyone on the Internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

You are creating a public repository in your personal account.

Create repository


3. Fork any public repository or your own from another account

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

 usernewncrd / new-fork

✔ new-fork is available.


By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

This project aims to develop an interactive blogging platform where users can create and engage with content

☒ Copy the `main` branch only

Contribute back to mriganko07/Blogging-Website-New by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

Create fork

4. Clone the forked repo:
git clone https://github.com/usernewncrd/git-practice.git
cd git-practice

```
ubuntu@ubuntu:~/newuser/git-demo$ git clone https://github.com/usernewncrd/new-fork
Cloning into 'new-fork'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (7/7), 28.85 KiB | 1.07 MiB/s, done.
ubuntu@ubuntu:~/newuser/git-demo$ cd new-fork/
ubuntu@ubuntu:~/newuser/git-demo/new-fork$
```

5. Create a branch:
git checkout -b feature

```
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git checkout -b feature
Switched to a new branch 'feature'
```

6. Make changes, then commit:
echo "Feature added" >> newfile.txt
git add .
git commit -m "Added new feature"

```
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ echo "Feature Added" >> newfile.txt
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git add .
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git commit -m "Added new feature"
[feature ec92d67] Added new feature
1 file changed, 1 insertion(+)
create mode 100644 newfile.txt
```

7. Merge branch into main:
git checkout master
git merge feature

```
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git merge feature
Updating d0bf9b1..ec92d67
Fast-forward
 newfile.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 newfile.txt
ubuntu@ubuntu:~/newuser/git-demo/new-fork$
```

8. Rebase branch (alternative to merge):
git checkout feature
git rebase master

```
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git checkout feature
Switched to branch 'feature'
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git rebase main
Current branch feature is up to date.
ubuntu@ubuntu:~/newuser/git-demo/new-fork$
```

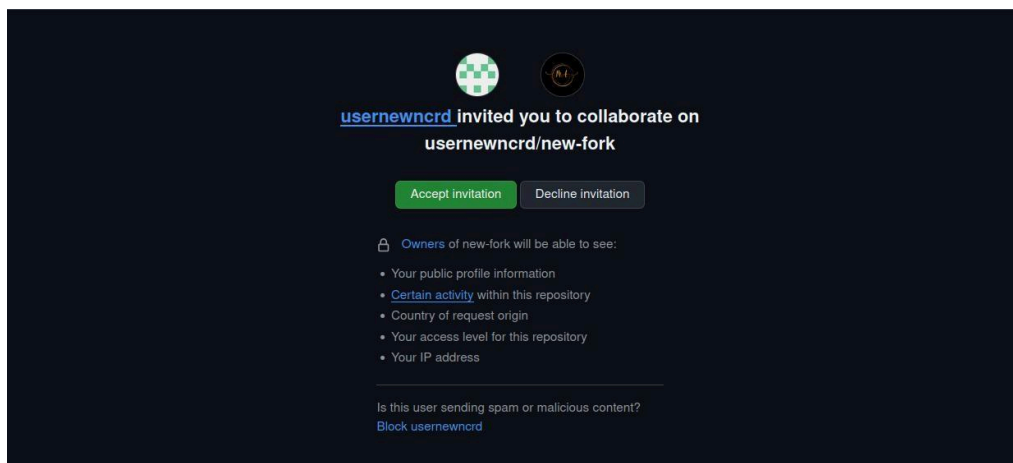
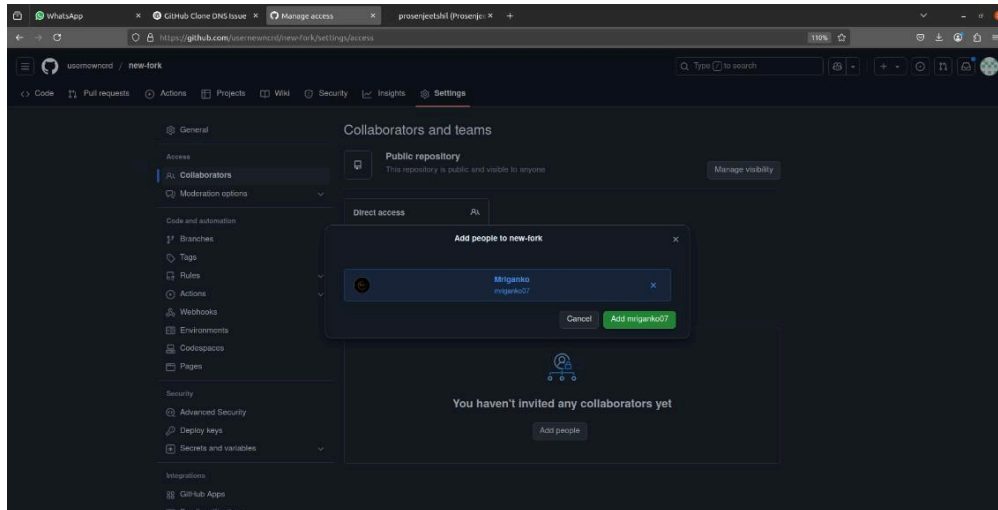
9. Push to GitHub:
git push origin feature

```
ubuntu@ubuntu:~/newuser/git-demo/new-fork$ git push origin feature
Username for 'https://github.com': usernewncrd
Password for 'https://usernewncrd@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 283.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/usernewncrd/new-fork/pull/new/feature
remote:
To https://github.com/usernewncrd/new-fork
 * [new branch]   feature -> feature
ubuntu@ubuntu:~/newuser/git-demo/new-fork$
```

PRACTICAL NO.3

Using Git for Collaboration

1. Using Git for Collaboration



2. Friend clones the repo:

```
git clone https://github.com/usernewncrd/git-practice.git  
cd team-repo  
git checkout -b bug-fix
```

```
ubuntu@ubuntu:~/newuser/git-demo$ git clone https://github.com/usernewncrd/git-practice.git  
Cloning into 'git-practice'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Unpacking objects: 100% (3/3), 1000.37 KiB | 2.44 MiB/s, done.  
ubuntu@ubuntu:~/newuser/git-demo$ cd team-repo  
bash: cd: team-repo: No such file or directory  
ubuntu@ubuntu:~/newuser/git-demo$ git checkout -b bug-fix  
Switched to a new branch 'bug-fix'  
ubuntu@ubuntu:~/newuser/git-demo$
```

3. Friend makes changes and pushes:

```
echo "Bug fixed" >> bug.txt
```

```
git add .
```

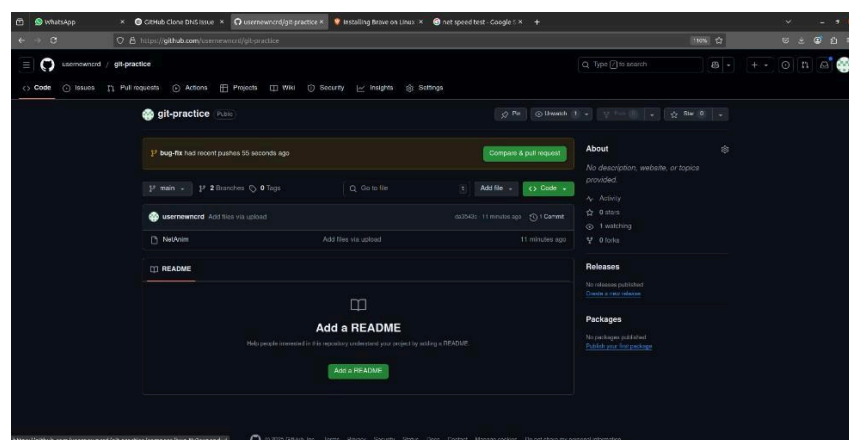
```
git commit -m "Fixed a bug"
```

```
ubuntu@ubuntu:~/newuser/git-demo$ echo "Bug fixed">>bug.txt
ubuntu@ubuntu:~/newuser/git-demo$ git add .
warning: adding embedded git repository: git-practice
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> git-practice
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached git-practice
hint:
hint: See "git help submodule" for more information.
warning: adding embedded git repository: new-fork
ubuntu@ubuntu:~/newuser/git-demo$ git commit -m "Fixed the bug"
[bug-fix a816be3] Fixed the bug
3 files changed, 3 insertions(+)
create mode 100644 bug.txt
create mode 160000 git-practice
create mode 160000 new-fork
ubuntu@ubuntu:~/newuser/git-demo$
```

4. git push origin bug-fix

```
ubuntu@ubuntu:~/newuser/git-demo$ git push origin bug-fix
Username for 'https://github.com': usernewncrd
Password for 'https://usernewncrd@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 549 bytes | 549.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'bug-fix' on GitHub by visiting:
remote:   https://github.com/usernewncrd/git-practice/pull/new/bug-fix
remote:
To https://github.com/usernewncrd/git-practice.git
 * [new branch]      bug-fix -> bug-fix
ubuntu@ubuntu:~/newuser/git-demo$
```

5. Pull Request



PRACTICAL NO.4

Collaborating and Cloning using GitHub

1. Clone a public repository:
git clone https://github.com/usernewncrd/git-practice.git

```
ubuntu@ubuntu:~/newuser/git-demo$ git clone https://github.com/usernewncrd/git-practice.git
Cloning into 'git-practice'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1000.37 KiB | 2.44 MiB/s, done.
```

2. Create a branch:
git checkout -b update-readme

```
ubuntu@ubuntu:~/newuser/git-demo$ git checkout -b update-readme
Switched to a new branch 'update-readme'
ubuntu@ubuntu:~/newuser/git-demo$
```

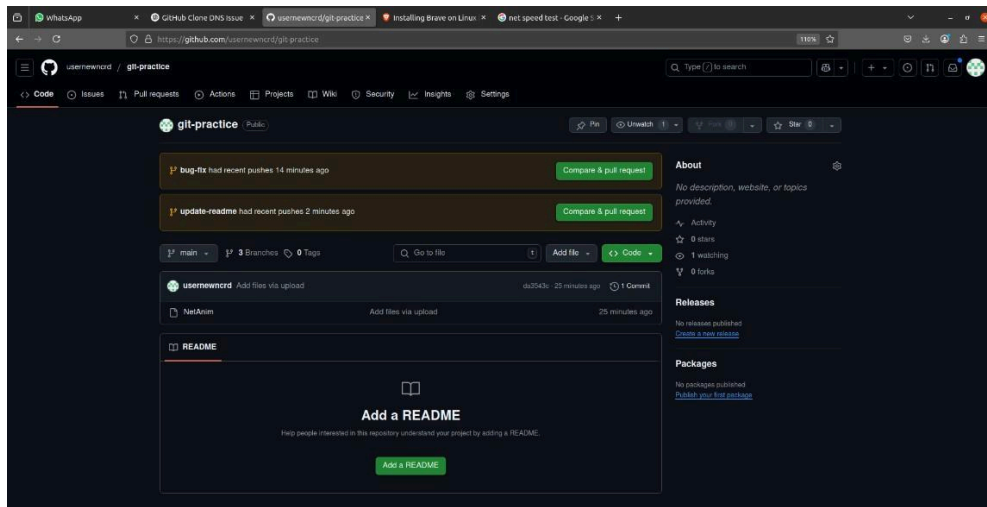
3. Edit and commit changes:
echo "Added a line" >>
README.md git add README.md
git commit -m "Updated README"

```
ubuntu@ubuntu:~/newuser/git-demo$ echo "Added a line">>README.md
ubuntu@ubuntu:~/newuser/git-demo$ git add README.md
ubuntu@ubuntu:~/newuser/git-demo$ git commit -m "Updated README"
[update-readme 11aa668] Updated README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
ubuntu@ubuntu:~/newuser/git-demo$
```

4. Push and open pull request:

```
ubuntu@ubuntu:~/newuser/git-demo$ git push origin update-readme
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 290 bytes | 290.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'update-readme' on GitHub by visiting:
remote:   https://github.com/usernewncrd/git-practice/pull/new/update-readme
remote:
To https://github.com/usernewncrd/git-practice.git
 * [new branch]      update-readme -> update-readme
ubuntu@ubuntu:~/newuser/git-demo$
```

5. git push origin update-readme

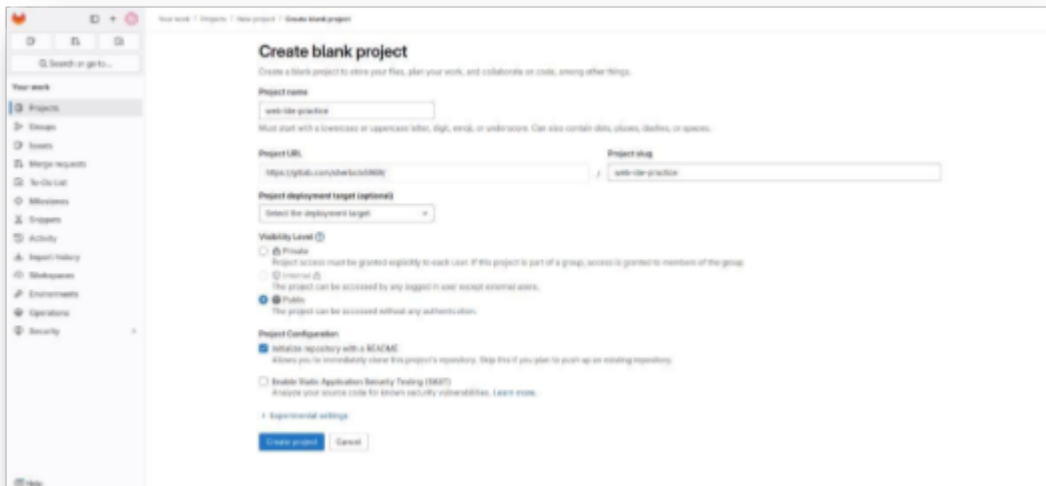


PRACTICAL NO.5

Using GitLab Web IDE

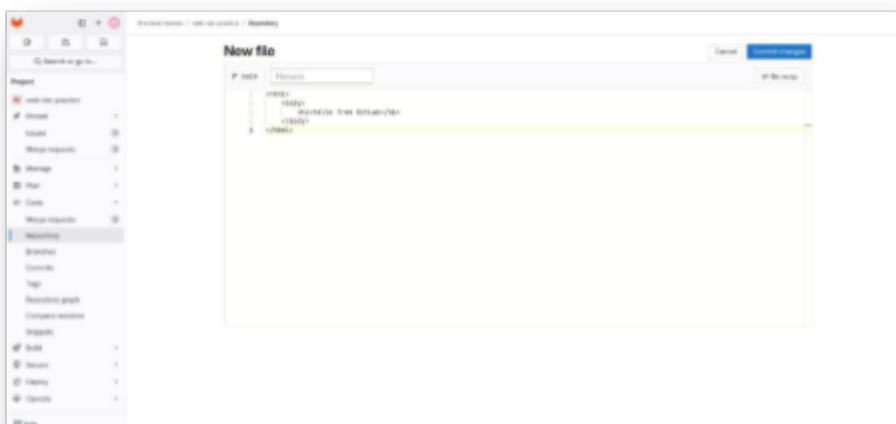
Steps:

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

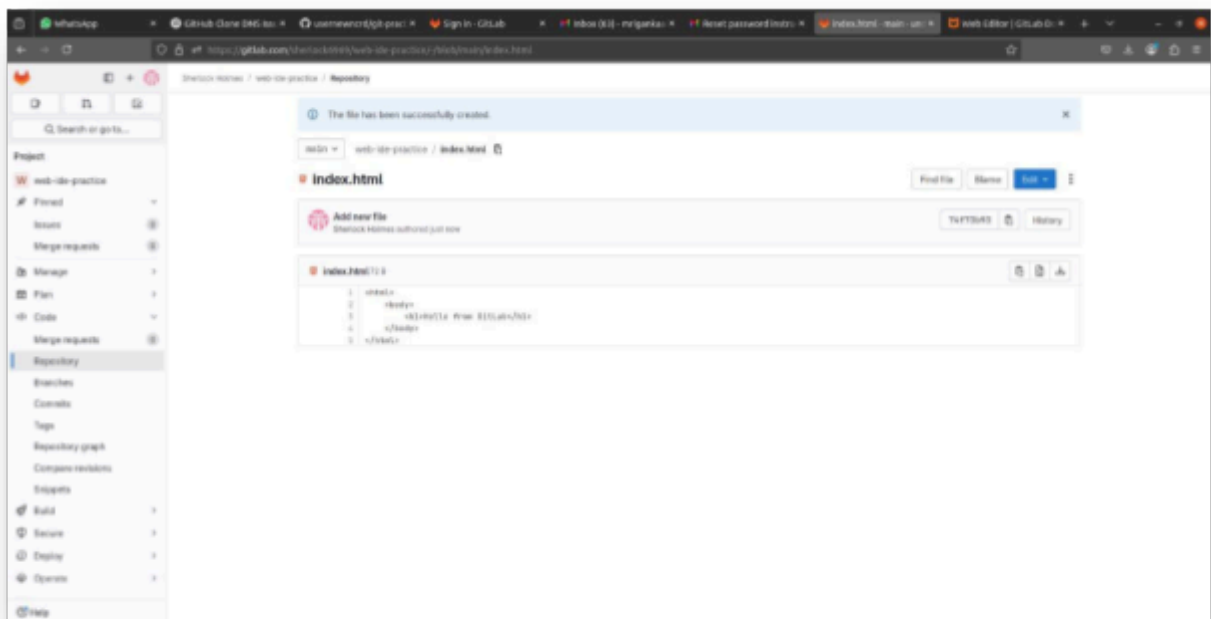
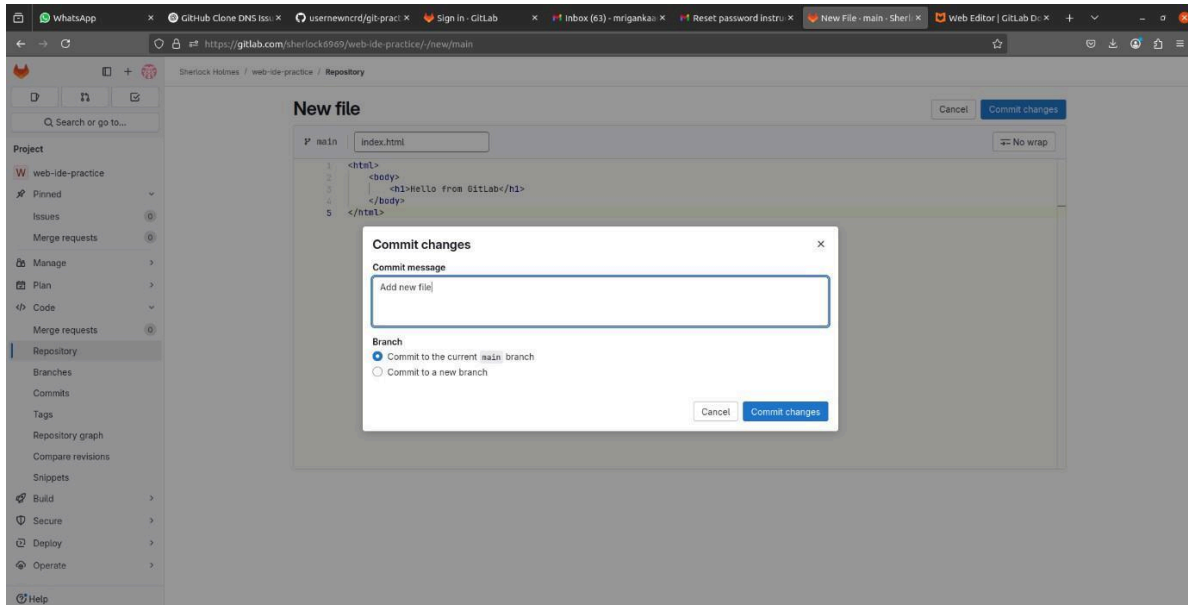


4. Create a file (index.html):

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



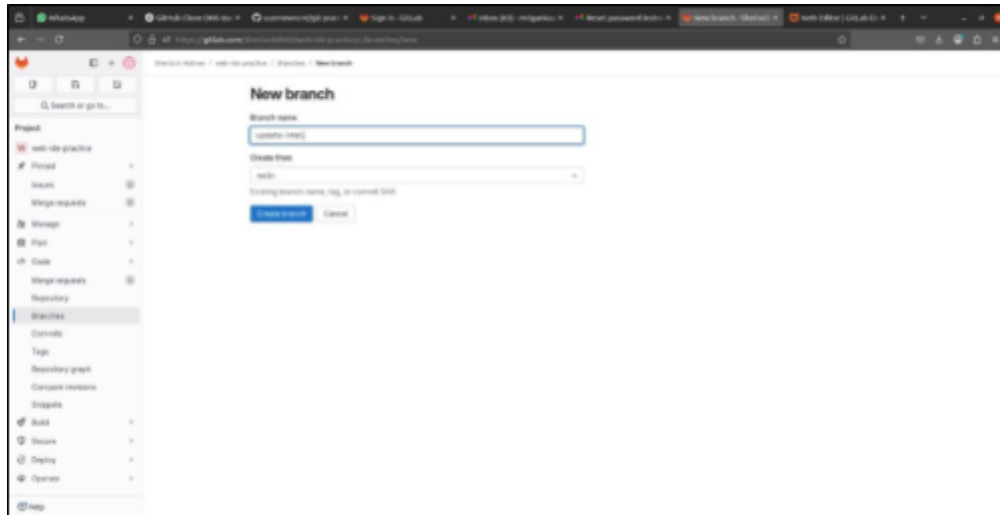
4. Click Commit and push changes.



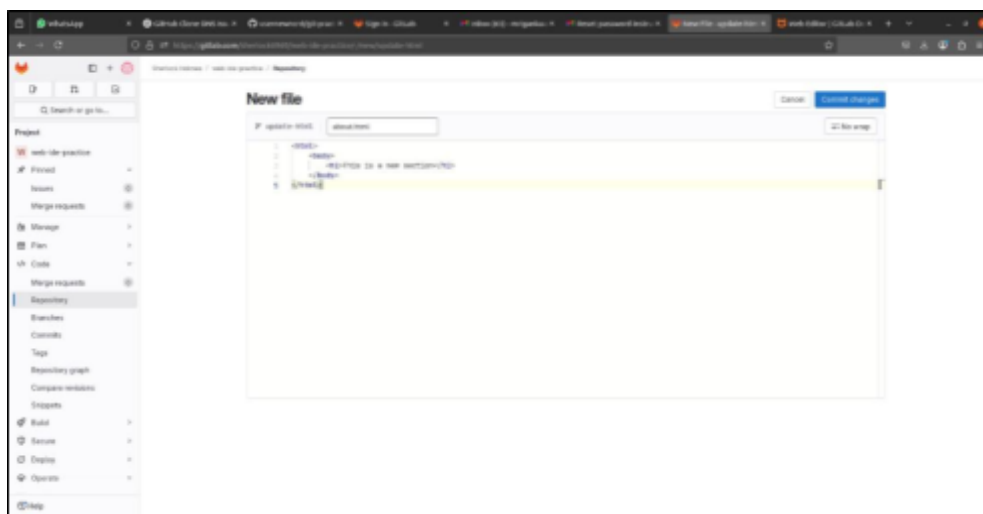
PRACTICAL NO.6

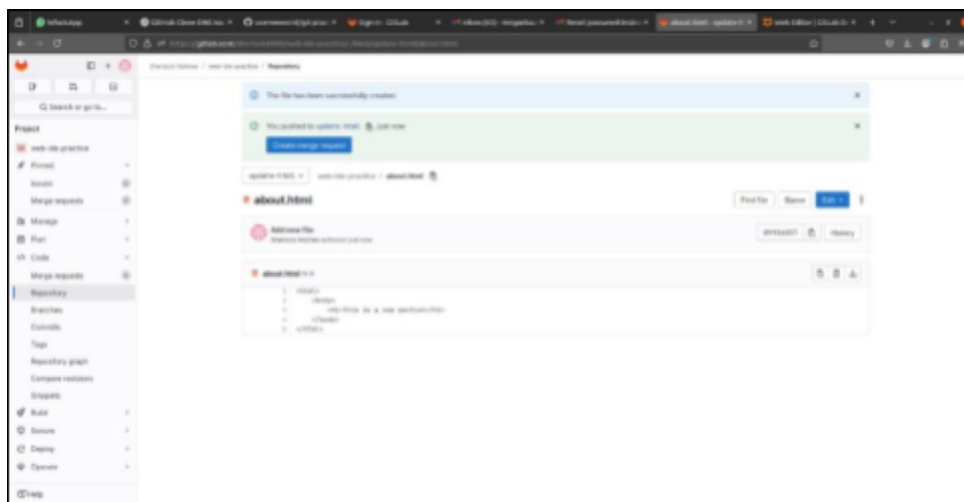
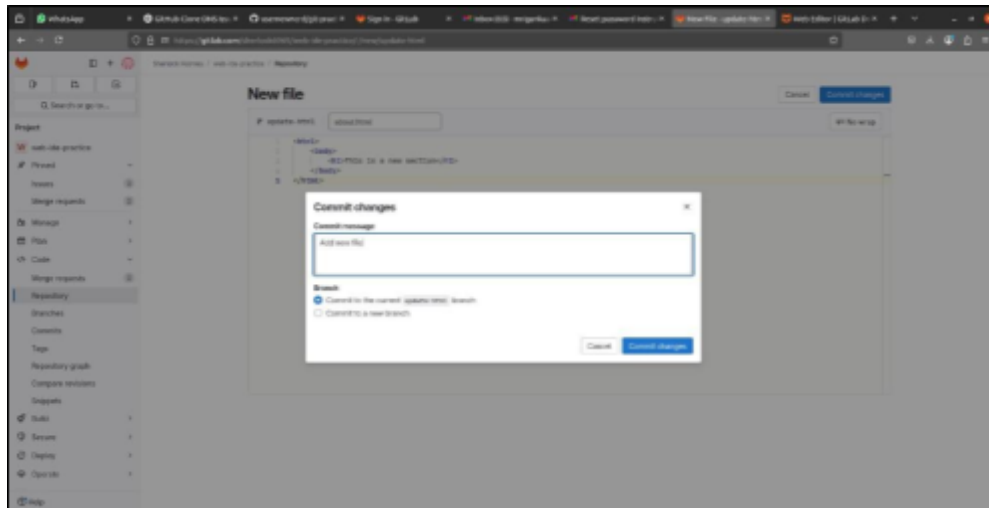
Performing merge requests using GitLab

1. Create a new branch in Web IDE.

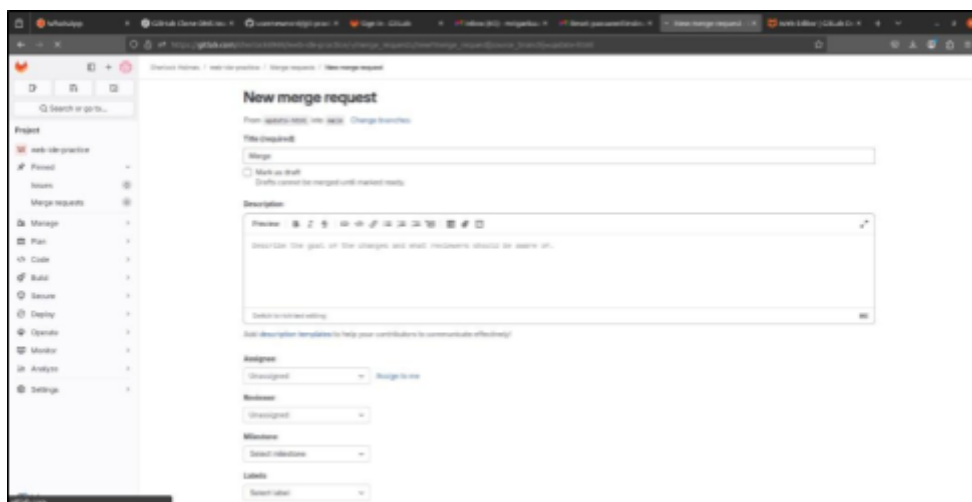


2. Add/edit a file and commit.

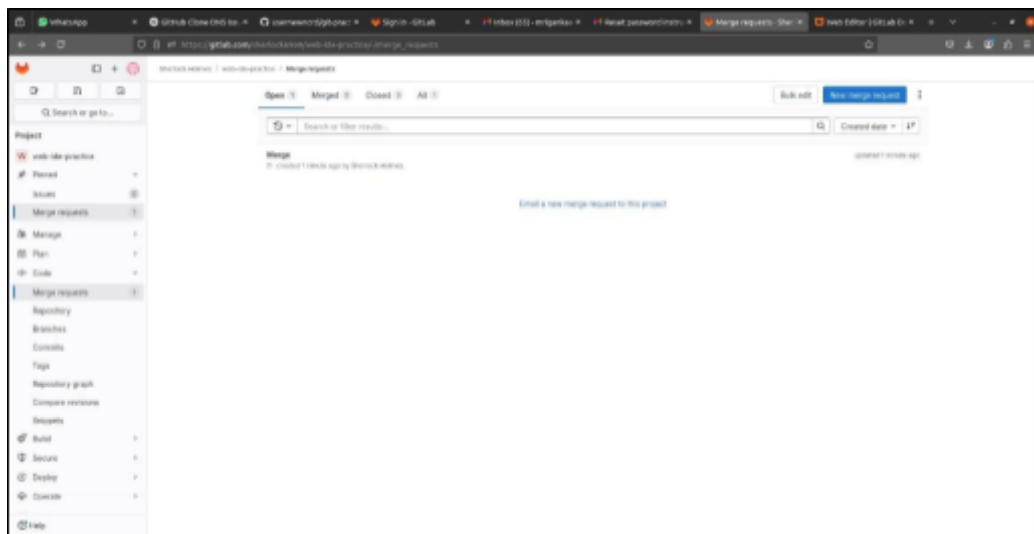




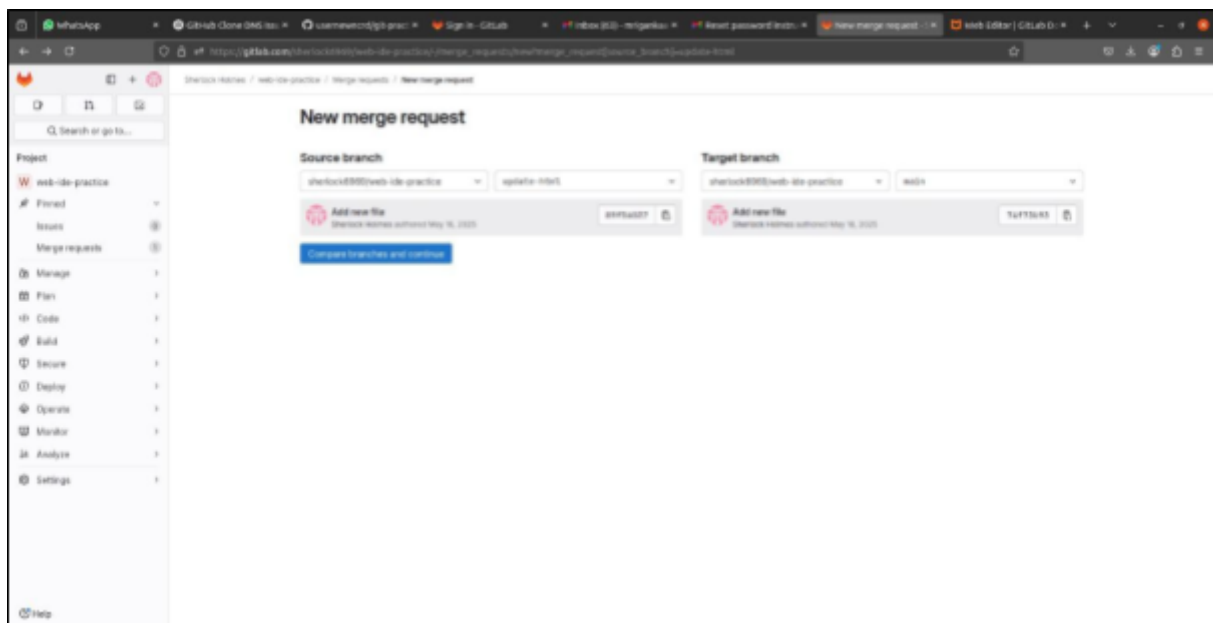
3. Click on Merge Requests > New Merge Request.



4. Select source and target branches.



5. Submit and merge after review.

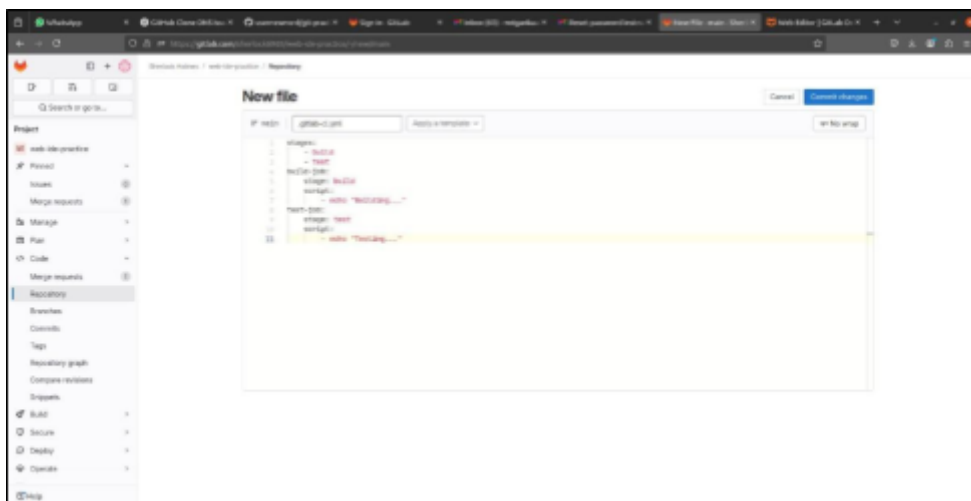


PRACTICAL NO.7

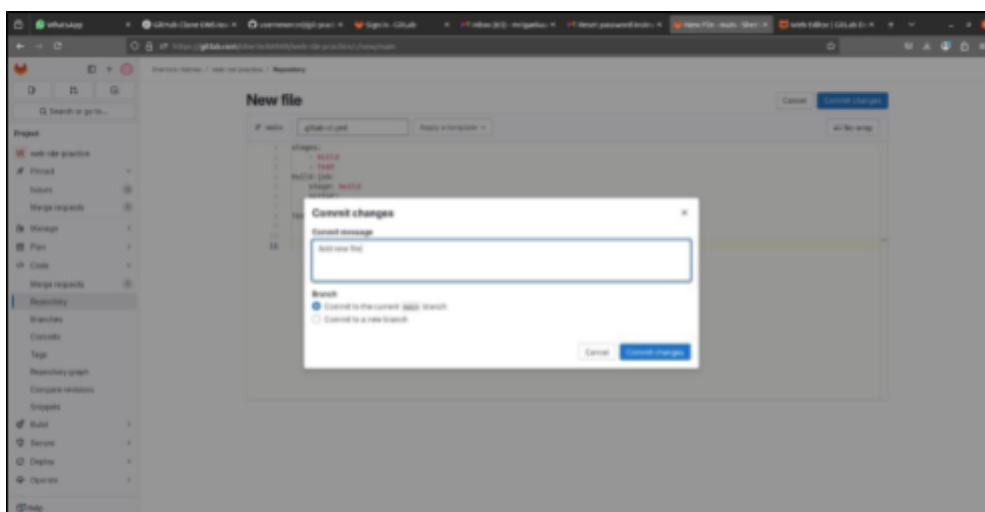
Workflow management in GitLab

Steps:

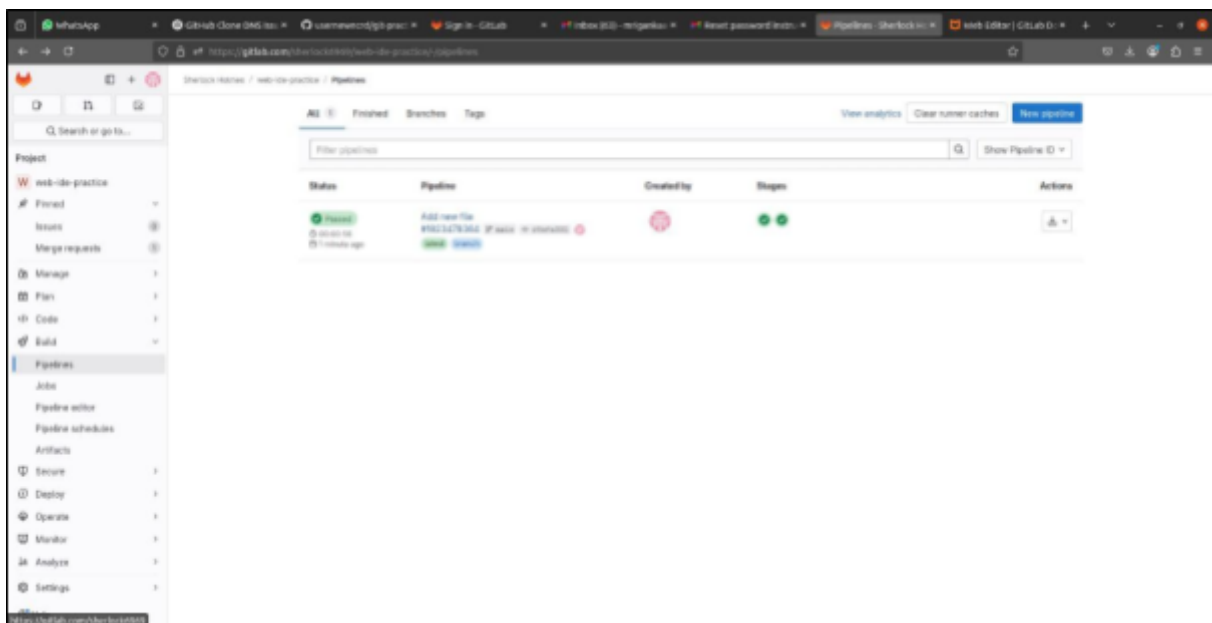
1. In your repo, create
.gitlab-ci.yml: stages:
 - build
 - testbuild-job:
 - stage: build
 - script:
 - echo"Building..."test-job:
 - stage: test
 - script:
 - echo "Testing..."



2. Commit and push.



3. Go to CI/CD > Pipelines and view the build/test stages.

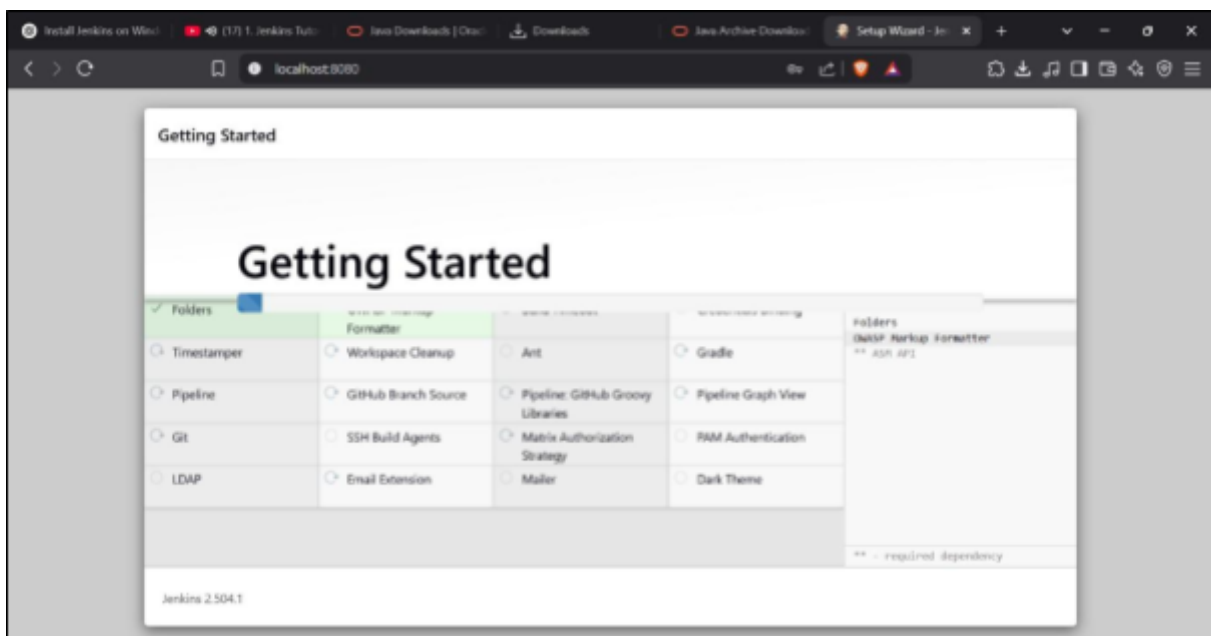
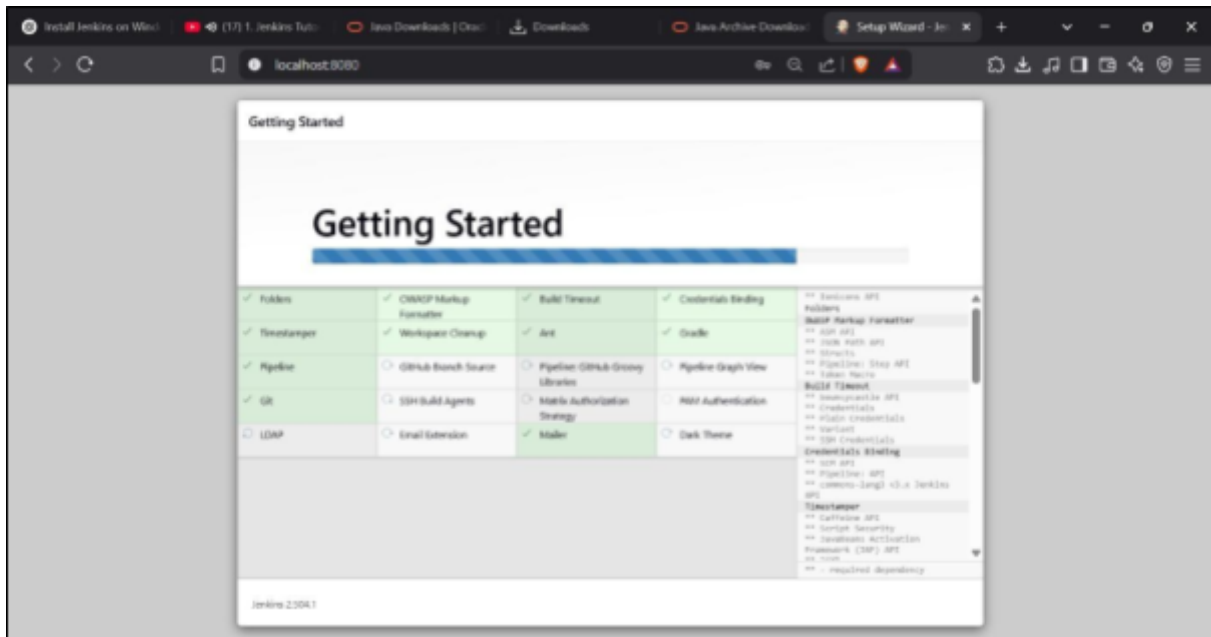


PRACTICAL NO.-8

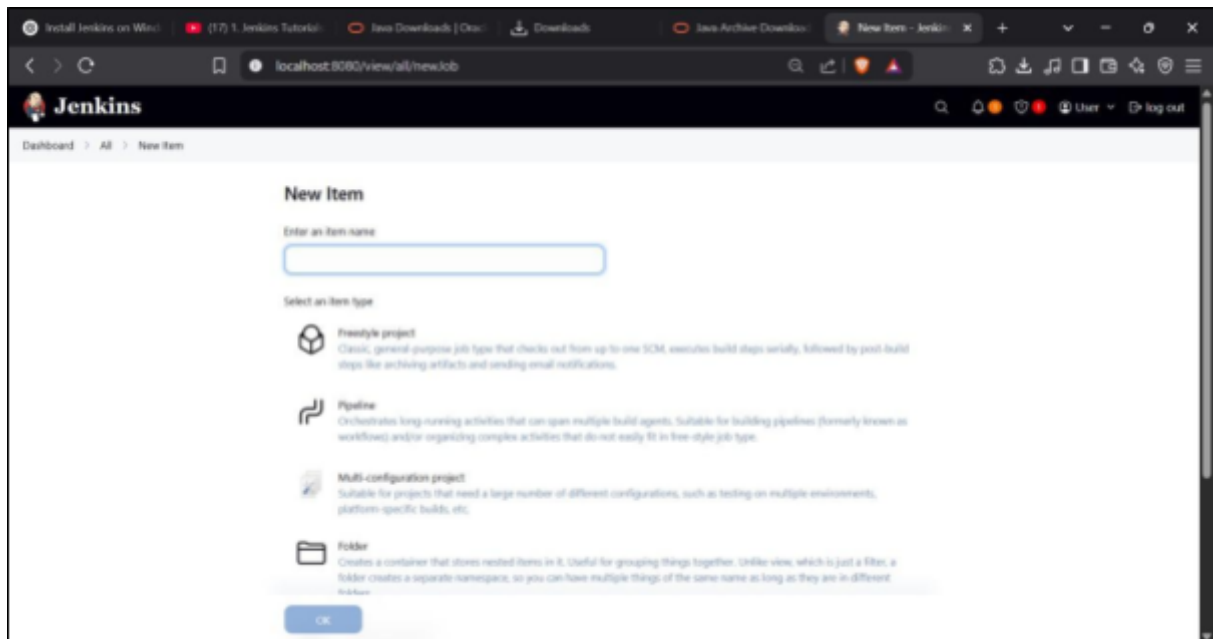
Demonstrate Continuous Integration and development using Jenkins

Steps:

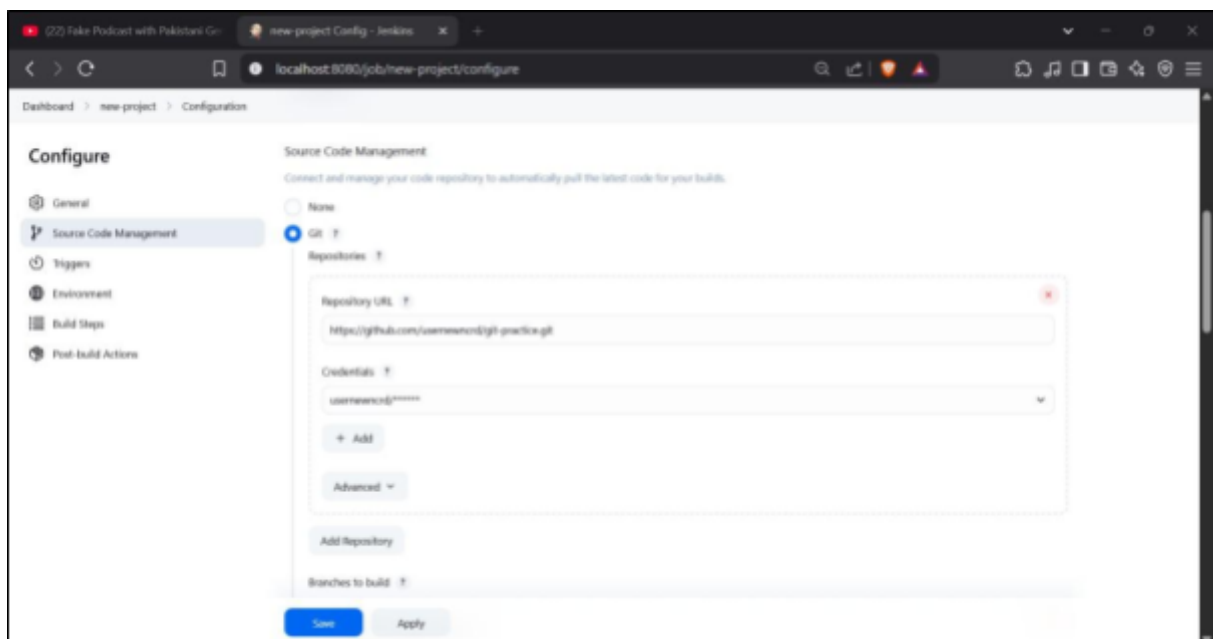
1. Install Jenkins (visit <https://www.jenkins.io>)
2. Run Jenkins: <http://localhost:8080>



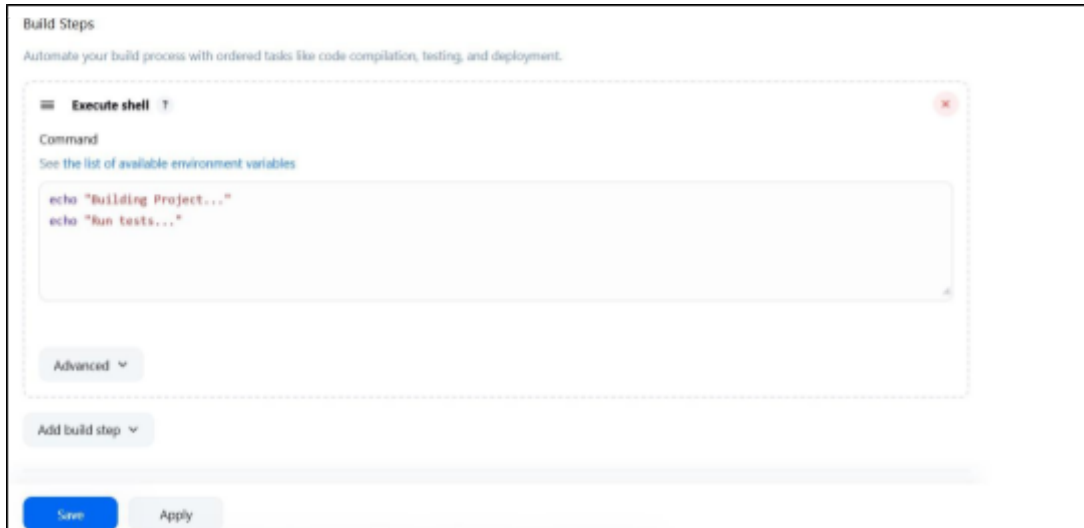
3. Create new Freestyle Project: CI-Demo



4. Under Source Code Management, choose Git and enter your repo URL.



5. Add Build Step > Execute
Shell: echo "Building
Project..."
echo "Run tests..."



6. Save and click Build Now.



7. Check output in Console Output.

```
+ echo 'Building Project...'
Building Project...
+ echo 'Run tests...'
Run tests...
Finished: SUCCESS
```

PRACTICAL NO.9

Explore docker commands for content management

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/  
lsstat /home/ubuntu/index.html: no such file or directory
```

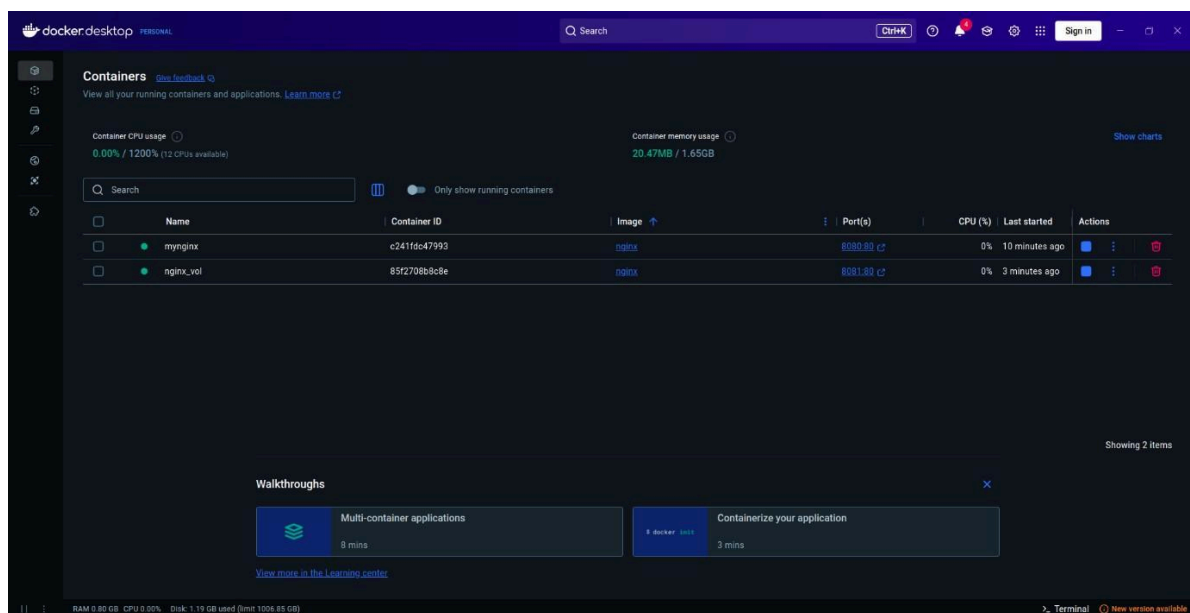
8. Create and use Docker volume for persistent content
docker volume create mydata
docker run -d -p 8081:80 --name nginx_vol -v mydata:/usr/share/nginx/html nginx
Now any data added to the /usr/share/nginx/html inside the container will persist even if the container is removed.

```
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 NO.10

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 :-



```
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`

```
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`

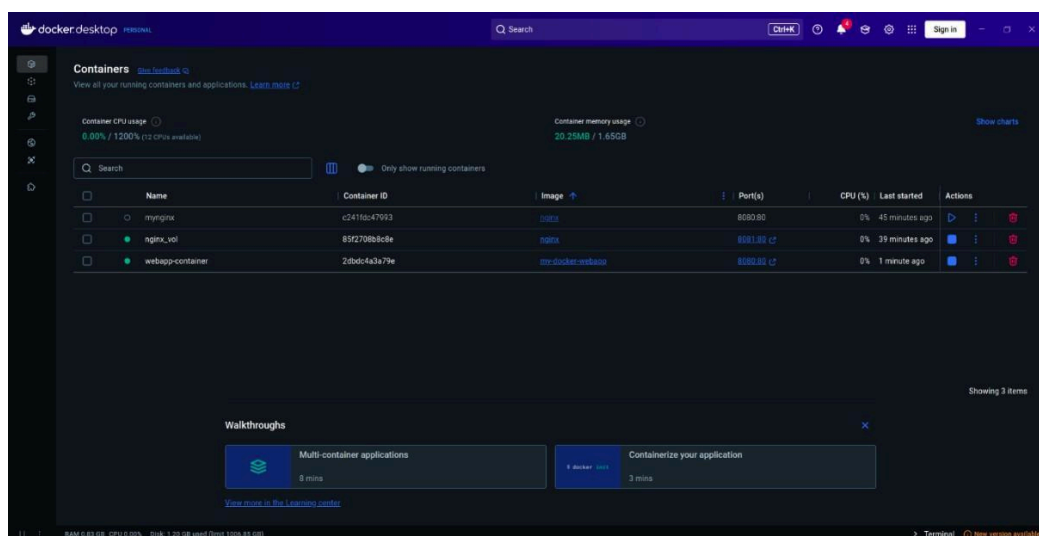
```
ubuntu@ubuntu:~/DevOps$ docker stop webapp-container
webapp-container
```

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 NO.11

Ad-hoc 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://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-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
  libwireshark13 libwsutil11 libxext-dev qt5-qmake qt5-qmake-bin qtbase5-dev qtbase5-dev-tools x11proto-xext-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 python3 libpython3 libwireshark13 vlc-plugin-qt libzmq5
  python2.7-minimal vlc-plugin-skins2 vlc-plugin-visualization vlc-lib0
  libgraph6 libpython2.7 python2.7 vlc-plugin-notify libvlc5 python3-lpython
  libpython2.7-dev libgvc6-plugins-gtk libpostproc55 liblab-gamut1 libvcore9
  libvlc-bin libpmix2 libzvb10 libavcodec58 vlc libcdt5 libavutil56 vlc-data
  libpathplan4 libavdevice58 libwscale5 libgvr2 libstdl2-2.0-0 libmysofa1
  inetutils-traceroute vlc-plugin-video-splitter libpython2.7-minimal
  libgraphviz-dev libpython5 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-5.15.0-139 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
  libmysqldclient12 libpoppler-cpp0v5 libpoppler-glib8 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
  libwireshark13 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:
  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-xmltodict
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-xmltodict
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.
Get: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 0.7.19-3ubuntu1 [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 [6,804 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-xmltodict 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"
}
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...Building dependency tree...Reading state information...The following packages were automatically installed and are no longer required:\n chroni
n-codescs-fmpop-extra gstreamer1.0-vaapi\n libgstreamer-plugins-bad1.0-0 libat5concurrent5 libat5opengl5-dev libat5sql5 libat5sql5-sqlite libat5tests libat5kan-dev libat5shark3 libwre
tap10\n libwst11 libxext-dev qt5-qmake qt5-qmake-bin qtbase5-dev\n qtbase5-dev-tools x11proto-text-dev\nUse 'sudo apt autoremove' to remove them.\n\nThe following additional packages will
be installed:\n libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter\n libnginx-mod-mail libnginx-mod-stream nginx-common nginx-core\n\nSuggested packages:\n fcgiwrap nginx-doc\n\nTh
e following NEW packages will be installed:\n libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter\n libnginx-mod-mail libnginx-mod-stream nginx-common nginx-core\n0 upgraded
, 7 newly installed, 0 to remove and 67 not upgraded.\nNeed to get 605 kB of archives.\nAfter this operation, 2141 kB of additional disk space will be used.\nGet:1 http://in.archive.ubuntu.c
on/ubuntu focal-updates/main amd64 nginx-common all 1.18.0-0ubuntu1.7 [37.8 kB]\nGet:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18
.0-0ubuntu1.7 [14.8 kB]\nGet: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]\nGet:4 http://in.archive.ubuntu.c
on/ubuntu focal-updates/main amd64 libnginx-mod-mail amd64 1.18.0-0ubuntu1.7 [43.0 kB]\nGet:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libnginx-mod-stream amd64 1.18.0-0ubu
ntu1.7 [67.3 kB]\nGet:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 nginx-core amd64 1.18.0-0ubuntu1.7 [425 kB]\nGet:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main
amd64 nginx all 1.18.0-0ubuntu1.7 [3620 B]\nPreconfiguring packages ...
\nFetched 605 kB in 7s (85.6 kB/s)\nSelecting previously unselected package nginx-common.\n(Reading database ...
\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(Rea
ding 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(Rea
ding 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 fi
les and directories currently installed.)\nPreparing to unpack .../0-nginx-common_1.18.0-0ubuntu1.7_all.deb ...
\nUnpacking nginx-common (1.18.0-0ubuntu1.7) ...
\nSelecting previously unselected package libnginx-mod-http-image-filter.\nPreparing to unpack .../1-libnginx-mod-http-image-filter_1.18.0-0ubuntu1.7_amd64.deb ...
\nUnpacking libnginx-mod-http-image-filter (1.18.0-0ubuntu1.7) ...
\nSelecting previously unselected package libnginx-mod-http-xslt-filter.\nPreparing to unpack .../2-libnginx-mod-http-xslt-filter_1.18.0-0ubuntu1.7_amd64.deb ...
\nUnpacking libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.7) ...
\nSelecting previously unselected package libnginx-mod-mail.\nPreparing to unpack .../3-libnginx-mod-mail_1.18.0-0ubuntu1.7_amd64.deb ...
\nUnpacking libnginx-mod-mail (1.18.0-0ubuntu1.7) ...
\nSelecting previously unselected package libnginx-mod-stream.\nPreparing to unpack .../4-libnginx-mod-stream_1.18.0-0ub
untu1.7_amd64.deb ...
\nUnpacking libnginx-mod-stream (1.18.0-0ubuntu1.7) ...
\nSelecting previously unselected package nginx-core.\nPreparing to unpack .../5-nginx-core_1.18.0-0ubuntu1
.7_amd64.deb ...
\nUnpacking nginx-core (1.18.0-0ubuntu1.7) ...
\nSelecting previously unselected package nginx.\nPreparing to unpack .../6-nginx_1.18.0-0ubuntu1.7_all.deb ...
\nUnpacking nginx (1.18.0-0ubuntu1.7) ...
\nSetting up libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.7) ...
\nSetting up libnginx-mod-mail (1.18.0-0ubuntu1.7) ...
\nSetting up libnginx-mod-http-image-filter (1.18.0-0ubu
ntu1.7) ...
\nSetting up libnginx-mod-stream (1.18.0-0ubuntu1.7) ...
\nSetting up nginx-core (1.18.0-0ubuntu1.7) ...
\nSetting up nginx (1.18.0-0ubuntu1.7) ...
\nProcessing triggers for man-db (2.9.1-1) ...
\nProcessing triggers for ufw (0.36-0ubuntu1.1) ...
\n",
    "stdout_lines": [
      "Reading package lists...",
      "Building dependency tree...",
      "Reading state information...",
      "The following packages were automatically installed and are no longer required:"
    ]
  }
}
```


PRACTICAL NO.12

Using 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

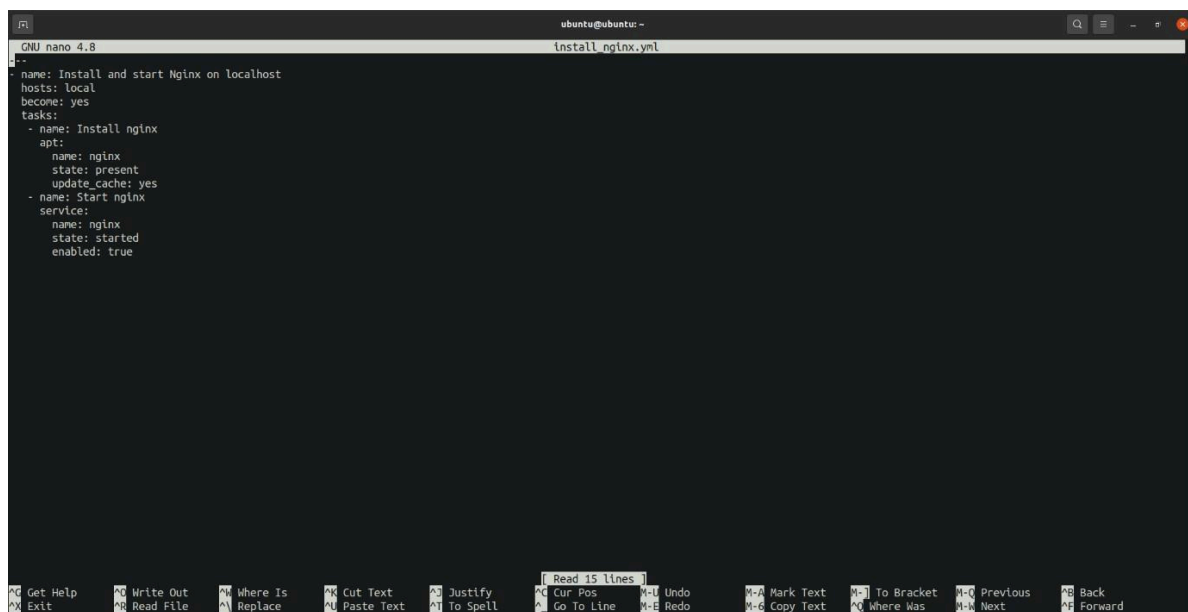
Nginx service:

name: nginx

state: started

enabled: true

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



```
ubuntu@ubuntu: ~
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
```

