GitHub

1) sudo apt install git
2) git --version
3) git config --global user.name "Vrushali"
4) git config --global user.email "vrushali@gmail.com"
5) git init myrepo (create repository)
   myrepo folder can be created into file manager.
6) cd myrepo (change directory)
7) list -a (display hide file)
8) touch sampal.txt (create new first text file into folder and write anything in that file)
9) git add sampal.txt (add all file data)
10) git add . (add all)
11) git status
12) git commit -m "first change"
13) git log (check history)
14) git branch "branch1" (create new branch)
15) git checkout "branch1" (checkout onto new branch)
16) touch newfile.txt (create new file into folder)
17) git add .
18) git commit -m "second change"
19) git status
20) git checkout master
21) git merge branch1

- Github
1) Login on Github
2) create a new repository on github
3) copy repository url from github
   Then goto terminal
4) git clone <url> (paste the repo url that is copied)
5) cd <reponame>
   cd repo1

6) git status
Goto file manager and check that repository folder is created or not. If folder is created. Then type some message into README.md file & save then goto Terminal.
7) git status
8) git add .
9) git commit -m "change3 done"
10) git branch "branch2"
11) git checkout branch2
12) git push origin <branch-name>
13) Goto Github :-
right click → settings → Developer setting → personal access token → token classic → generate new token → generate new token classic → click checkbox (except workflow) → generate token → copy id of token.
Goto Terminal :-
14) git remote set-url origin http://paste token id@github.com/paste(username/reponame)
15) git push origin <branch-name>
16) Goto Github :
github → pull request option → New pull request → compare → <branch-name> → create pull request → Merge pull request → confirm merge
17) git status
If we want to make changes on main branch then add some content on README.md file & save it.
18) git checkout main
19) git add .
20) git commit -m "change done"

21) git push origin main
22) git fetch origin main
23) git pull origin main
24) git add .
25) git commit -m " changes done "
26) git push origin main

git remote set-url origin https://github.com/Vrushali1210/Cake_shop_website.git

GitLab

(project name :- Sampal)

1) gitlab → account create → create project.
Open gitlab → create a new project on gitlab → add some content in it → Copy project Url.
Goto Terminal →

2. git clone <gitlab_project_url>
3. cd sampal
4. ls
5. git branch branch1
6. git checkout branch1
7. ls
8. touch demo.txt
9. ls
10. git checkout master.
11. ls
12. git checkout branch1
13. git add .
14. git commit -m "first commit"
15. ls
16. git checkout master
17. ls
18. git push origin branch1
19. git checkout master
20. ls
21. git pull origin master
22. ls.

Create personal access token.

Bitbucket

1. Goto bitbucket → login → create repository → click on clone (copy the URL) → paste the url on Terminal → Password.
2. How to create password :-
   Setting → Personal Bitbucket Settings → App password → create app password → copy id → paste on Terminal (suppose, reponame ⇒ demo)
3. Goto Terminal → paste the Url
4. cd demo
5. touch sampal.txt (add some content in file)
6. git add.
7. git commit -m "first commit"
8. git status
9. ls
10. git branch branch1
11. git checkout branch1
12. git status
13. ls
14. touch sampal2.txt
15. git add.
16. git commit -m "second commit"
17. git status
18. ls
19. git push origin branch1.
20. Goto bitbucket → go to repo → create pull request → merge.
21. git pull origin main

Jenkins.

- Terminal

① sudo systemctl status jenkins

- New Terminal

② ngrok http 8080
Here we see the status of ngrok → from that click on a ~~for~~ forwarding link and open a link on browser / or copy the link ~~upto~~ upto free.app and paste it on a browser and open it.

③ Click on visit site

④ username ⇒ kkwagh
password ⇒ srv001

- Github

⑤ login on a Github (on new window) ⇒ Open the repository that we want to build on a jenkins ⇒ Copy the repo / project link from code (http).

- Jenkins

⑥ Click on new item ⇒ Enter item name (eg. de ⇒ select freestyle Project ⇒ Click OK ⇒ from Source code Management ⇒ Select git ⇒ Paste the Git repo Url (that copied from github ⇒ From Branches to Build (write main instead of master) ⇒ from Build Triggers select GitHub hook trigger for GITScm polling ⇒ Click Apply Save.

- GitHub.

⑦ On repo ⇒ Click setting ⇒ Click Webhooks ⇒ Add webhook ⇒ On payload URL (copy the URL from jenkins or Terminal upto free.app and paste here) (on this URL add a ~~git~~ /github-webhook/ after free.app) ⇒ Select Content type (application/json) ⇒ select send me everything ⇒ Click Add Webhook ⇒ then Refresh (see green tick) ⇒ Click on Build Now (after successful build) ⇒ Click Workspace and run the html file ⇒ Optional ↓ (found error)

⑧ Click config ⇒ Build steps ⇒ select Execute shell ⇒ Write command ⇒ cp -r */var/www/html/ ⇒ apply & save ⇒ Click Build Now.

o.o.o.o.

Docker

1) sudo apt install docker.io
2) sudo apt get update
3) sudo systemctl enable docker
4) sudo systemctl status docker
   (create one folder eg:- myproject)
5) create one HTML(index.html) file.
   (add some content in it)
6) touch Dockerfile
   (Open that Dockerfile and add in it) ⇒

```
FROM nginx
COPY index.html /usr/share/nginx/html
```

7) sudo docker build -t myproject .
   (folder_name (that we ~~ha~~create))
8) sudo docker images
9) sudo docker run -d -p 8081:80 myproject
10) Goto browser.
    0.0.0.0:8081 ⇒ enter