1. Create a git repository and put java file with some program like hello world with demo.java rule: class name and file name should be same
2. class demo
3. {
4. public static void main(String []args)
5. {
6. System.out.println("My First Java Program.");
7. }
8. };
10. Go to jenkins <https://www.jenkins.io/doc/book/installing/linux/> and choose linux and LTS realses and copy paste each line by line in terminal
11. Run “sudo systemctl start jenkins”
12. Run “sudo systemctl status jenkins” if active then fine
13. Run ‘localhost:8080’ in browser a page will appear in that a path will there paste that in terminal using ‘sudo cat path’ and a passcode will be displayed on the terminal
14. Copy that passcode and paste it in that website of localhost:8080 and move forward and select option install suggest plugins and click next
15. Chose ( user as admin and password of system ) u should skip so it will default choose that and if u reach dashboard then success
16. Then choose new item enter name and select free style project and click ok
17. Redirects to other page leave description as mpty and select github project and paste url of github
18. Choose souce code management as git and again paste repo url and down enter git credentials username and password
19. Change branch specifier to \*/main
20. In triggers choose build periodically and i schedule put \* \* \* \* \*
21. In build step choose execute shell and in command write
22. “javac demo.java
23. Java demo”
24. Save and in build click on that and choose console op and verify

Docker:

If push and pull then run ‘docker login’

Username:vj2909 pass:Vijay@2003

1)create a app folder using mkdir app

2) cd app

3)touch app.py

4)now open app.py and write following code

From flask import Flask

app=Flask(\_name\_\_)

@app.route(‘/’)

def main():

Return ‘hello from flask app’

If \_\_name\_\_==’\_\_main\_\_’:

app.run(host=‘0.0.0.0’)

5)again go terminal and got app folder in terminal and run touch Dockerfile

6)now open dockerfile and write following

FROM python:3.10

WORKDIR /app

COPY . /app

RUN pip install flask or pip install -r requirements.txt (\* optional if u create a requirements file)

EXPOSE 5000

CMD [“python3”,”app.py”]

7)run the “docker build -t fapp2 .”

8)if success ok else error then it shows a path in error like var/run/docker.sock then do:

cd

cd /var

ls

sudo chmod 666 docker.sock

cd

cd Desktop

cd app

Error resolves

If no push pull:

9)docker build -t fapp2 .

10))docker run -p 5000:5000 fapp2

If push pull

docker build -t vj2909/fapp2

docker run -p 5000:5000 vj2909/fapp2

docker push vj2909/fapp2

Go to docker hub and username and pass enter

docker pull vj2909/fapp2

Ssh:

1)sudo apt install openssh-server

2)ssh-keygen or ssh-keygen -t rsa -b 2048

3)put 2 times enter and in overwrite give ‘y’

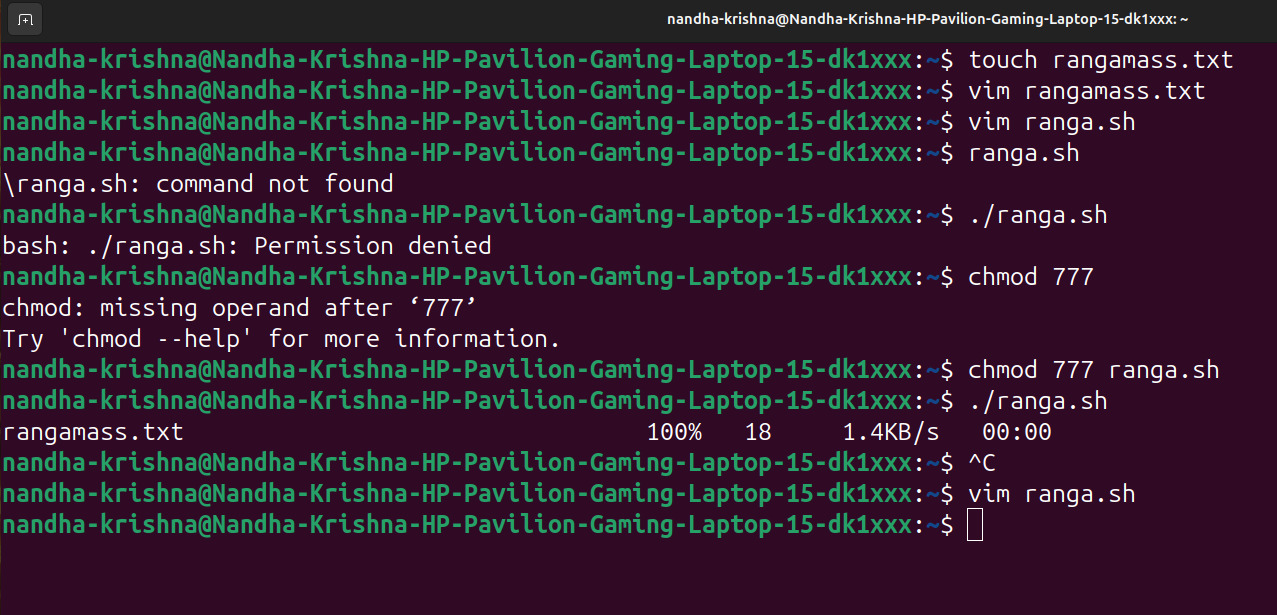
4)run ‘ifconfig’ on other system to get ip address and it will there in wlo:1 - inet

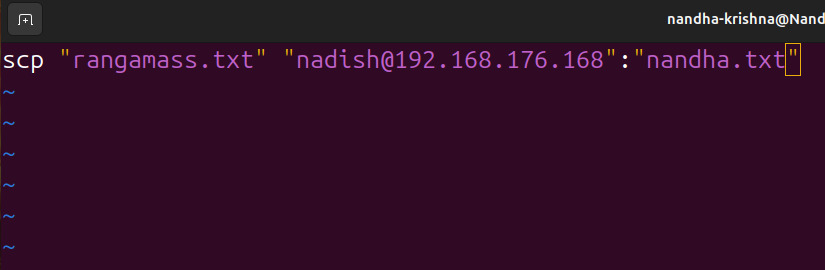
5)run who i am to get name of the other

6)ssh-copy-id {username}@{ip addr of other}

7)asks password of other system and enter it

8)ssh {username}@{ip add}

Ssh-file transfer:  




Pipe line:

Choose pipe line

In definition choose pipeline script from scm

An scm will open put git password anol

Change brach specifier to main

In script paths choose jenkins.txt avlo tha vera ethum pannathinga

Jenkins.txt:

pipeline {

agent any

stages {

stage('version') {

steps {

echo 'Stage Version'

}

}

stage('hello') {

steps {

echo 'Stage Hello'

sh 'javac Sort.java'

sh 'java Sort'

}

}

}

}

Test.java:

class Test

{

public static void main(String []args)

{

System.out.println("Hello World!!");

}

};

Build and run

Git conflict:

sudo apt install git

Git init

Create a code file

eg;Calc.java

public class Calc{

public static void main(String[] args) {

// Hardcoded inputs

double num1 = 25;

double num2 = 5;

char operation = '+'; // Change this to '-', '\*', or '/' for other operations

double result = 0;

// Perform the operation

switch (operation) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

if (num2 != 0) {

result = num1 / num2;

} else {

System.out.println("Error: Division by zero is not allowed.");

return;

}

break;

default:

System.out.println("Error: Invalid operation.");

return;

}

// Print the result

System.out.println("The result of " + num1 + " " + operation + " " + num2 + " is: " + result);

}

}

In vs code open terminal and do following:

Create a branch :git checkout -b main

Save it : git add .

Commit : git commit -m "main"

Create 2 branches :

Create a branch:git checkout -b a

Make changes to file

git add .

git commit -m "a"

go to back main : git checkout main

Create a branch:git checkout -b b

Change file

Merge with a :

git merge a

To change permissions : sudo chmod +x sci.sh

Or sudo chmod 777 sci.sh

Docker using jenkin : run the following command

If ur lazy go to chapter 2 by skipping chapter 1

Chapter 1:

mkdir app

cd app

git init

Git clone {go to github and copy the link ends with .git and paste here eg : jenny.git} jenny is repo name

cd jenny

git add .

git commit -m “name”

git push -u origin main

Asks user name and password enter

If success then ok and check files are added in git or not

If not :

Chapter 2:

Then manually add the files

go to jenkins follow same process mentioned for jenkins

In execute shell write this commanda

docker build -t fapp2 .

docker run -p 5000:5000 fapp2

For ssh remote server jenkins :

<https://chatgpt.com/share/673f6e37-24bc-800f-b2a5-73005dbb671f>

Create a containerized REST api application using Docker

https://chatgpt.com/share/673f6f49-a46c-800f-a22f-7b5a70db3f61