**9th and 10th questions (push, pull, merge)**

git clone <link>

Cd <repository name>

git branch <branch name>

git checkout <branch name>

vim <any name>

[first type escape :wq]

git add <any name>

git commit -m “type any message”

git push origin <give the name which is in the ( ) brackets

**Docker**

mkdir Vishnu

cd Vishnu

echo .>index.html

code .

echo .>dockerfile

code .

docker build -t Vishnu .

docker images

docker run -d -p --name Vishnu Vishnu

docker run -d -p 8080:80 –name Vishnu Vishnu

**HTML CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>My Dockerized Website</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

margin-top: 50px;

}

h1 {

color: #2c3e50;

}

</style>

</head>

<body>

<h1>Welcome to My Dockerized Website!</h1>

<p>This website is served using NGINX in a Docker container.</p>

</body>

</html>

**DOCKER FILE:**

# Use the official NGINX image as a base image

FROM nginx:alpine

# Copy the HTML file into the NGINX default directory

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

# Expose port 80 to allow access to the web server

EXPOSE 80

# Start the NGINX server

CMD ["nginx", "-g", "daemon off;"]

**Docker with kubernets**

mkdir Vishnu

cd Vishnu

echo .>app.py

code .

echo .>dockerfile

code .

echo .>requirements.txt

code .

docker build -t Vishnu/flask-api:latest .

app.py

from flask import Flask, jsonify

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return jsonify(message="Welcome to the Flask API!")

@app.route('/api/data', methods=['GET'])

def get\_data():

return jsonify(data={"key1": "value1", "key2": "value2"})

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

dockerfile

# Use the official Python image

FROM python:3.9-slim

# Set the working directory

WORKDIR /app

# Copy the application code

COPY app.py /app

# Install Flask

RUN pip install flask

# Expose port 5000

EXPOSE 5000

# Run the application

CMD ["python", "app.py"]

requirement.txt

flask

**normal kubernets**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.21

ports:

- containerPort: 80

kubectl apply -f nginx-deployment.yaml

kubectl get pods

kubectl expose deployment nginx-deployment --type=NodePort --name=nginx-service

kubectl get svc

go to chrom and type the number you get in x place

localhost:xxxxx

managing and verifying

kubectl scale deployment nginx-deployment --replicas=4

kubectl get pods

delete

kubectl delete svc nginx-service

kubectl delete deployment nginx-deployment

**normal jenkins**

docker pull jenkins/jenkins:lts

docker run -d -p 8080:8080 -p 50000:50000 --name jenkins -v jenkins\_home:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkins/jenkins:lts

docker ps

go to chrome

localhost:8080