Sap id :- 70612300027

Q1) You are a team leader and are explaning to your team how kubernetes can be useful for managing and organizing the companies website even in terms of high load or incoming requests. Demonstrate using kubernets software any one application of Kubernetes on the system. Further create a flowchart with labelled steps for explaining to the team how kubernetes will make it easier t deploy, manage and scale the company website without worrying the underlying infrastructure. Also explain any technical terms and their meaning related to Kubernetes deployment.

Answer

Created a simple python website to demonstrate Kubernetes

Create a python website:

App.py

# app.py

from flask import Flask

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

@app.route('/')

def home():

    return "Welcome to the Company Website!"

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host='0.0.0.0', port=80)

Created a docker file to create a image of the website

# Use the official Python base image from Docker Hub

FROM python:3.9-slim

# Set the working directory inside the container

WORKDIR /app

# Copy the Flask application code into the container

COPY app.py .

# Install the required dependencies (Flask)

RUN pip install Flask

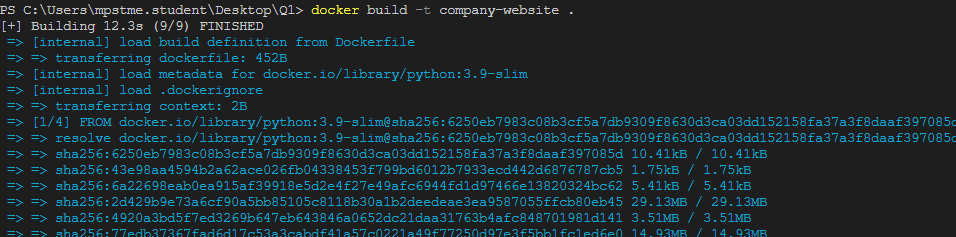
# Expose port 80 (since Flask runs on port 80 in our app)

EXPOSE 80

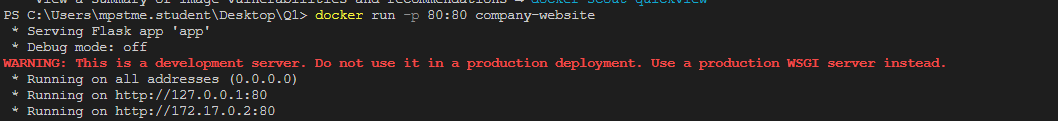
# Command to run the Flask app

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

Built an docker image of that website



Run the docker image





Deploying it to the Kubernetes

Deployment.yaml

# website-deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: company-website-deployment

spec:

  replicas: 3  # Number of replicas (pods) to run

  selector:

    matchLabels:

      app: company-website

  template:

    metadata:

      labels:

        app: company-website

    spec:

      containers:

      - name: company-website-container

        image: company-website:latest  # Replace with your Docker image URL

        ports:

        - containerPort: 80

Service.yaml

# website-service.yaml

apiVersion: v1

kind: Service

metadata:

  name: company-website-service

spec:

  selector:

    app: company-website

  ports:

    - protocol: TCP

      port: 80      # Exposes port 80 to the outside world

      targetPort: 80 # The port inside the container

  type: LoadBalancer  # Exposes the service externally (use NodePort for local clusters)

Applying the commands





Kubernets cluster wasn’t working in the pc.

Technical terms related to Kubernetes deployment

Key Kubernetes Concepts for Your Team:

Pods are the basic unit for running containers in Kubernetes.

Services help expose containers to the outside world and can automatically balance the traffic.

Deployments manage the pods and can automatically scale and roll out new versions of applications.

Horizontal Pod Autoscaler (HPA) automatically adjusts the number of pods based on metrics like CPU usage.

ReplicaSets maintain the specified number of replicas to ensure the website is always running.