Santhosh P – 22CSR177 III CSE C

Day 3 – Minikube installation and mysql

**Kubernetes**

Kubernetes (K8s) is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It helps in efficiently managing multiple containers across a cluster of machines, ensuring high availability, load balancing, and self-healing capabilities. Kubernetes is widely used for cloud-native applications and microservices architectures.

**Minikube**

Minikube is a lightweight Kubernetes implementation that runs a single-node Kubernetes cluster on a local machine. It is primarily used for development and testing purposes, allowing developers to experiment with Kubernetes features without needing a full-scale cluster. Minikube supports various container runtimes and can be installed on Windows, macOS, and Linux

curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

minikube start

minikube start

minikube status

YML file

ersion: '3'

services:

web:

image: nginx:latest

ports:

- 80:80

db:

image: mysql:latest

environment:

- MYSQL\_ROOT\_PASSWORD=secret

docker exec -it david-db-1 /bin/bash

mysql -u root -p

Docker compose:

**Docker Compose**

Docker Compose is a tool that allows you to define and manage multi-container Docker applications using a YAML configuration file (docker-compose.yml). It simplifies the process of running multiple interdependent services (such as a web server, database, and caching system) with a single command.

**Key Features:**

* **Multi-Container Management** – Define multiple services in one file.
* **Service Dependencies** – Automatically starts services in the correct order.
* **Networking** – Easily creates a shared network for containers.
* **Scalability** – Scale services up or down with a single command.

**Example docker-compose.yml:**

yaml

Copy code

version: '3'

services:

web:

image: nginx

ports:

- "8080:80"

db:

image: mysql

environment:

MYSQL\_ROOT\_PASSWORD: example

**Usage:**

sh

Copy code

# Start all services

docker compose up -d

# Stop and remove containers

docker compose down

Docker compose commands:

# Start and run containers in the background

docker compose up -d

# Start containers in the foreground (logs will be shown)

docker compose up

# Stop containers

docker compose down

# Restart containers

docker compose restart

# View running containers

docker compose ps

# View logs of services

docker compose logs

# View logs of a specific service

docker compose logs <service\_name>

# Build or rebuild services

docker compose build

# Stop containers without removing them

docker compose stop

# Start stopped containers

docker compose start

# Execute a command in a running container

docker compose exec <service\_name> <command>

# Remove stopped containers, networks, and volumes

docker compose down --volumes

# Show configuration details

docker compose config

# Scale a service (e.g., run 3 instances of a service)

docker compose up --scale <service\_name>=3 -d

Pipeline code

pipeline {

agent any

tools {maven "maven"}

stages {

stage('SCM') {

steps {

git branch: 'master', url: 'https://github.com/Santhosh-P-2005/DevOps.git'

}

}

stage('Build') {

steps {

sh 'mvn clean package'

}

}

stage('build to images') {

steps {

script {

sh 'docker build -t santhosh9405/webapp1 .'

}

}

}

stage('push to hub') {

steps {

script {

withDockerRegistry(credentialsId: 'docker\_cred', toolName: 'docker', url: 'https://index.docker.io/v1/') {

sh 'docker push santhosh9405/webapp1'

}

}

}

}

}

}

