SARAN AVINASH B – 22CSR179 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

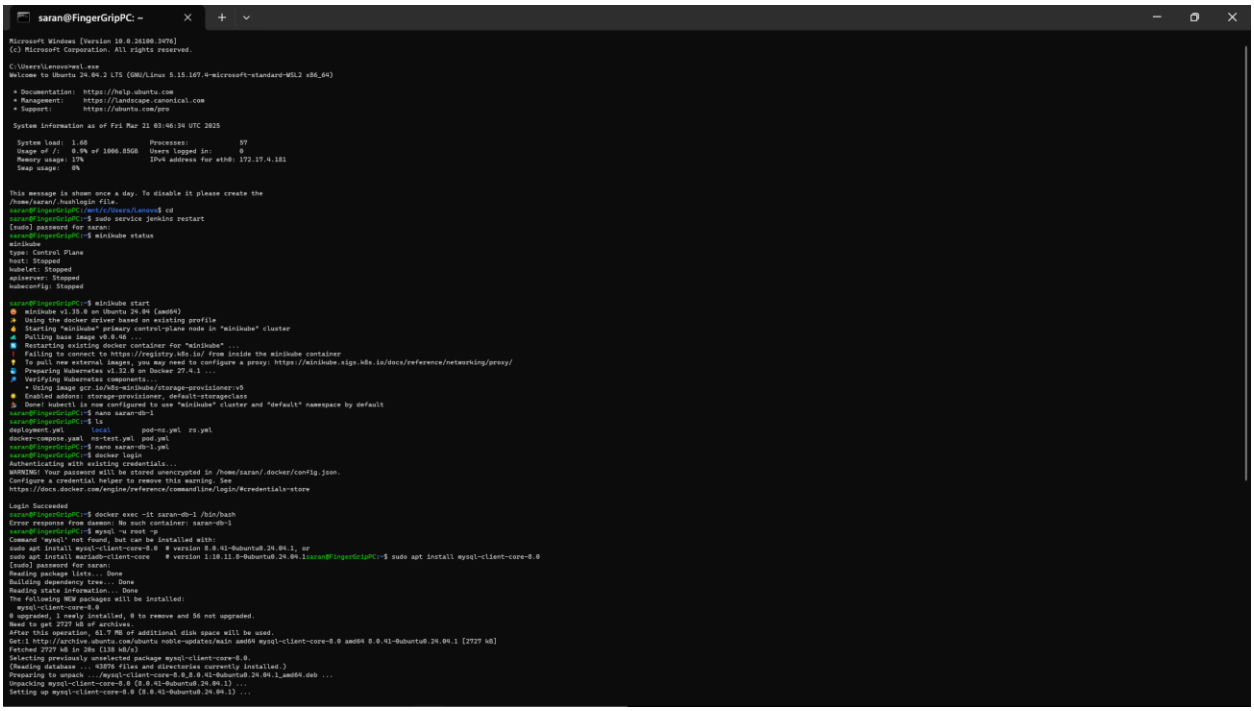
docker compose start

```
# 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/Saran-Avinash/DevOps.git'
         }
      }
      stage('Build') {
         steps {
            sh 'mvn clean package'
```

```
        }
    }
    stage('build to images') {
        steps {
            script {
                sh 'docker build -t saranavinashb/webapp1 .'
            }
        }
    }
    stage('push to hub') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'docker_cred', toolName: 'docker', url:
'https://index.docker.io/v1/') {
                        sh 'docker push saranavinashb/webapp1'
                }
            }
        }
    }

  }
}
```


saran@FingerGripPC: ~
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Lenovo>wsl.exe
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Fri Mar 21 03:46:34 UTC 2025
System load: 1.68 Processes: 97
Usage of /: 8.9% of 1006.85GB Users logged in: 0
Memory usage: 17% IPv4 address for eth0: 172.17.4.181
Swap usage: 0%
This message is shown once a day. To disable it please create the
/home/saran/.hushlogin file.
saran@FingerGripPC:/mnt/c/Users/Lenovo$ cd
saran@FingerGripPC:~$ sudo service jenkins restart
[sudo] password for saran:
saran@FingerGripPC:~$ minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped
saran@FingerGripPC:~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Restarting existing docker container for "minikube" ...
Failing to connect to https://registry.k8s.io/ from inside the minikube container
To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
saran@FingerGripPC:~$ nano saran-db-1
saran@FingerGripPC:~$ ls
deployment.yml local pod-ns.yml rs.yml
docker-compose.yaml ns-test.yml pod.yml
saran@FingerGripPC:~$ nano saran-db-1.yml
saran@FingerGripPC:~$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/saran/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
saran@FingerGripPC:~$ docker exec -it saran-db-1 /bin/bash
Error response from daemon: No such container: saran-db-1
saran@FingerGripPC:~$ mysql -u root -p
Command 'mysql' not found, but can be installed with:
sudo apt install mysql-client-core-8.0 # version 8.0.41-0ubuntu0.24.04.1, or
sudo apt install mariadb-client-core # version 1:10.11.8-0ubuntu0.24.04.1saran@FingerGripPC:~$ sudo apt install mysql-client-core-8.0
[sudo] password for saran:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
mysql-client-core-8.0
0 upgraded, 1 newly installed, 0 to remove and 56 not upgraded.
Need to get 2727 kB of archives.
After this operation, 61.7 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.41-0ubuntu0.24.04.1 [2727 kB]
Fetched 2727 kB in 20s (138 kB/s)
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 63876 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.41-0ubuntu0.24.04.1_amd64.deb ...
Unpacking mysql-client-core-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Setting up mysql-client-core-8.0 (8.0.41-0ubuntu0.24.04.1) ...