Rahul Siddharth D H – 22CSR158

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-linuxamd64 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

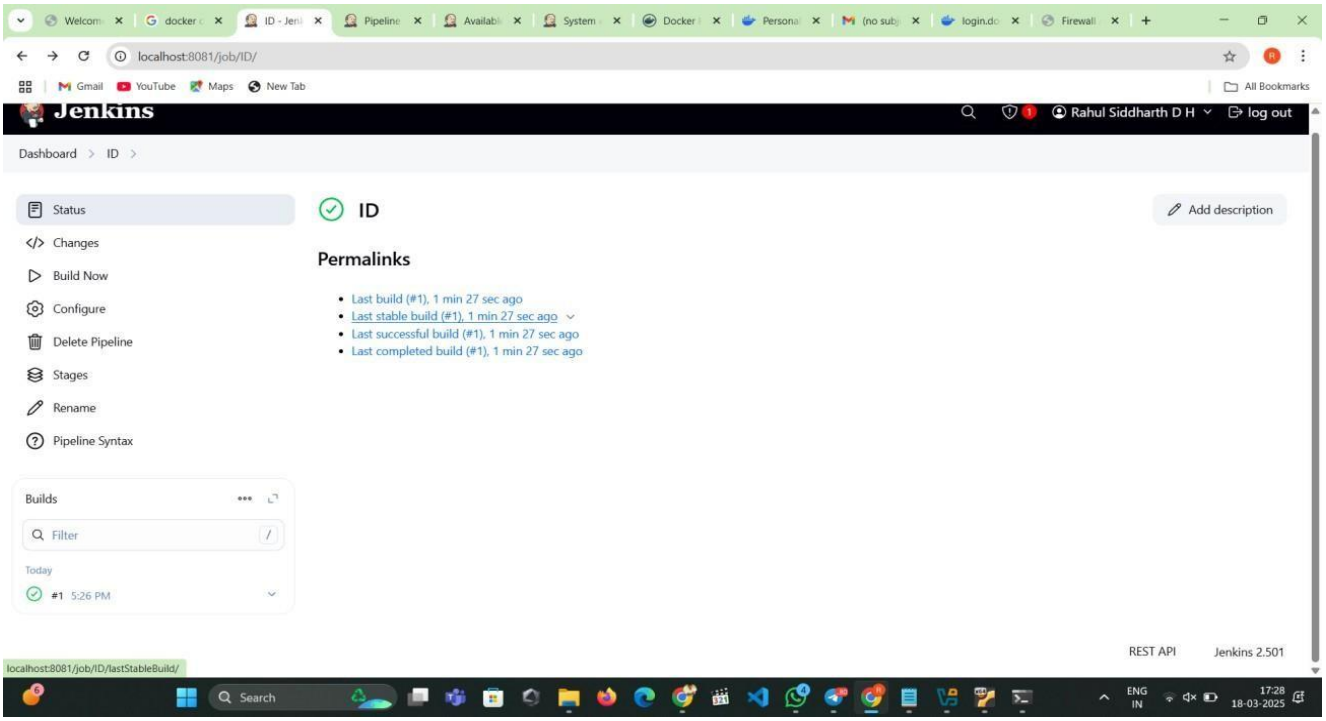
compose start

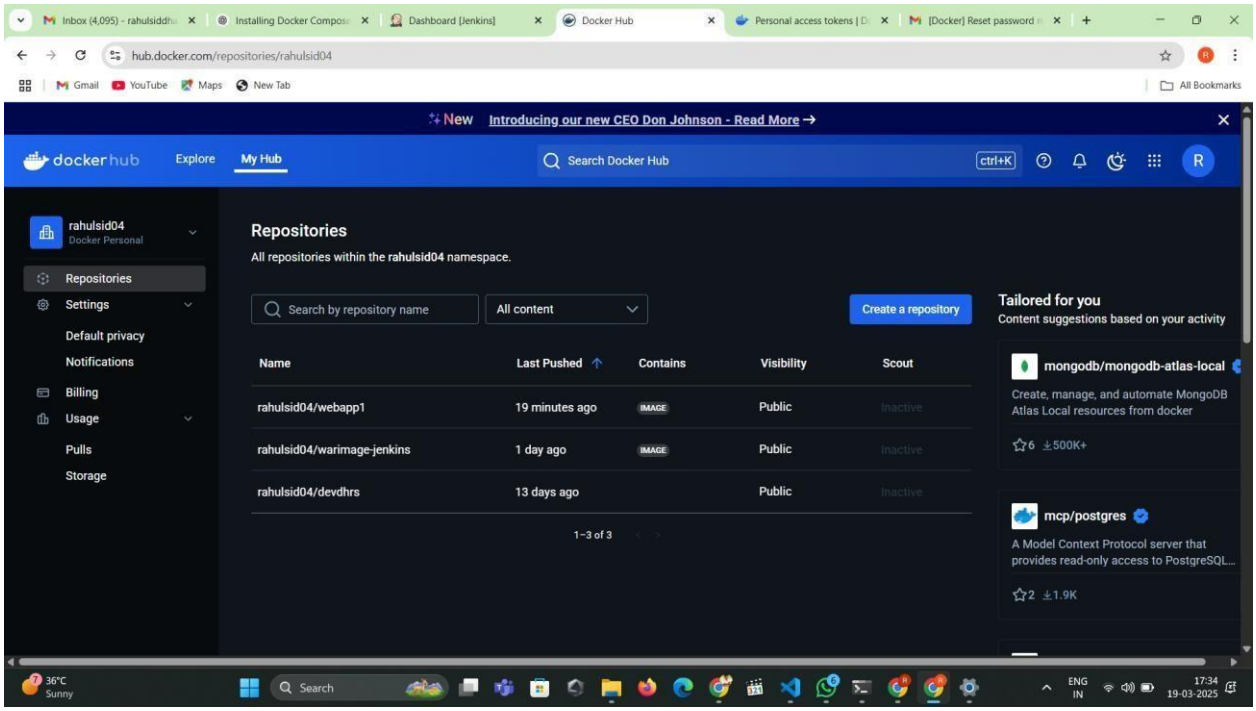
Execute a command in a running container docker

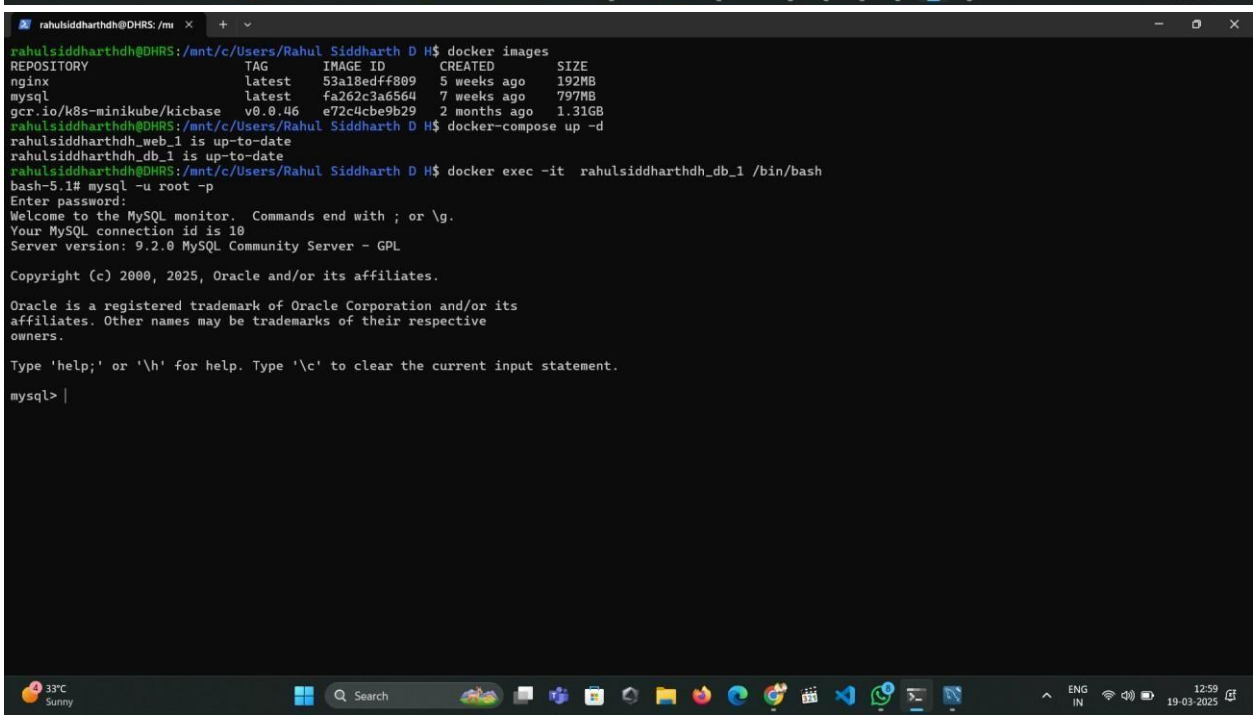
compose exec <service_name> <command>

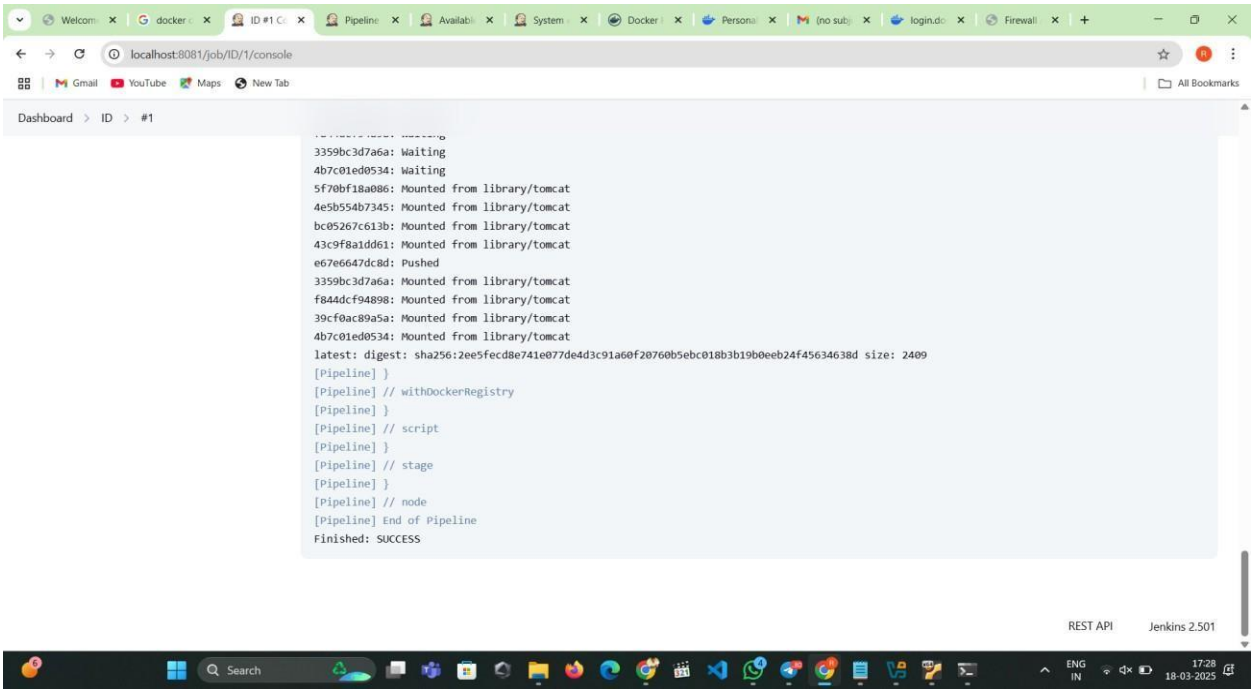
```
# Remove stopped containers, networks, and volumes docker
compose down --volumes

# 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 {
```

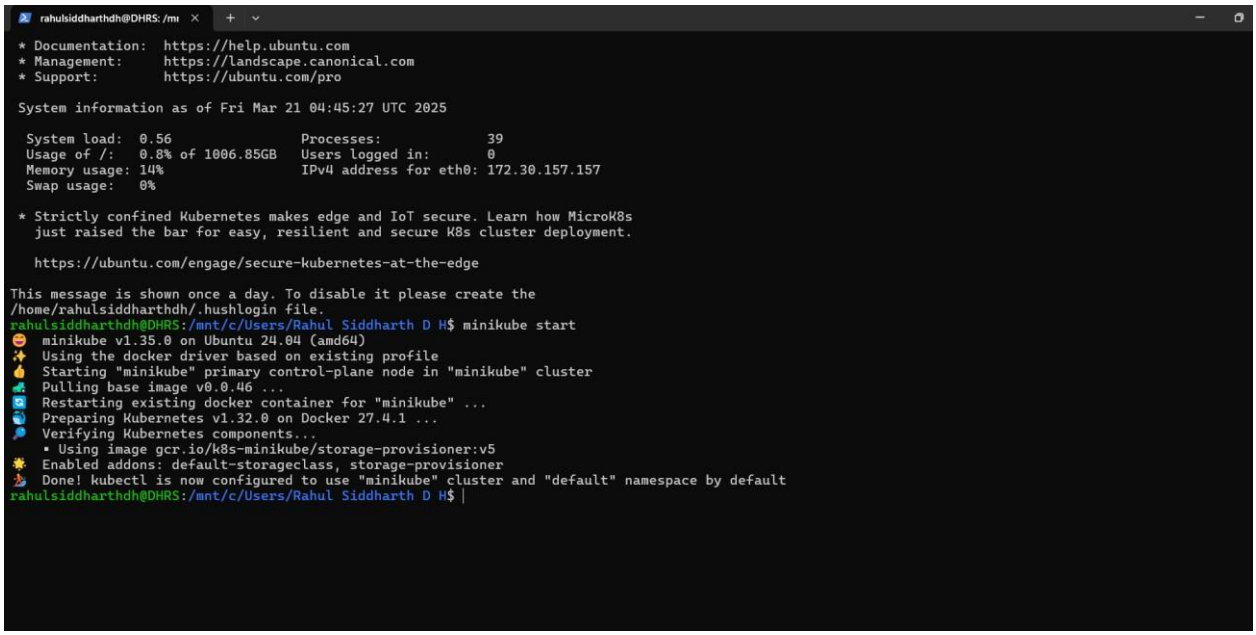
```
                sh 'docker build -t saranavinashb/webapp1 .'
            }
        }
    }
    stage('push to hub') {
        steps {
script {
                withDockerRegistry(credentialsId: 'docker_cred', toolName: 'docker', url:
'https://index.docker.io/v1/') {                    sh 'docker
push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

  }
}
```


localhost:8081/job/ID/
Gmail
YouTube
Maps
New Tab
All Bookmarks
Jenkins
Rahul Siddharth D H
log out
Dashboard
ID
Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
ID
Add description
Permalinks
Last build (#1), 1 min 27 sec ago
Last stable build (#1), 1 min 27 sec ago
Last successful build (#1), 1 min 27 sec ago
Last completed build (#1), 1 min 27 sec ago
Builds
Filter
Today
#1 5:26 PM
REST API
Jenkins 2.501
localhost:8081/job/ID/lastStableBuild/
Search
ENG
IN
17:28
18-03-2025


New Introducing our new CEO Don Johnson - Read More →
docker hub
Explore
My Hub
Search Docker Hub
ctrl+K
rahulsid04
Docker Personal
Repositories
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Storage
Repositories
All repositories within the rahulsid04 namespace.
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
rahulsid04/webapp1
19 minutes ago
IMAGE
Public
rahulsid04/warimage-jenkins
1 day ago
IMAGE
Public
rahulsid04/devdhrs
13 days ago
Public
1–3 of 3
Tailored for you
Content suggestions based on your activity
mongodb/mongodb-atlas-local
Create, manage, and automate MongoDB Atlas Local resources from docker
6 500K+
mcp/postgres
A Model Context Protocol server that provides read-only access to PostgreSQL...
2 1.9K


rahulsiddharthdh@DHRS:/mnt/c/Users/Rahul Siddharth D H$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 53a18edff809 5 weeks ago 192MB
mysql latest fa262c3a6564 7 weeks ago 797MB
gcr.io/k8s-minikube/kicbase v0.0.46 e72c4cbe9b29 2 months ago 1.31GB
rahulsiddharthdh@DHRS:/mnt/c/Users/Rahul Siddharth D H$ docker-compose up -d
rahulsiddharthdh_web_1 is up-to-date
rahulsiddharthdh_db_1 is up-to-date
rahulsiddharthdh@DHRS:/mnt/c/Users/Rahul Siddharth D H$ docker exec -it rahulsiddharthdh_db_1 /bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 9.2.0 MySQL Community Server - GPL
Copyright (c) 2000, 2025, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>


localhost:8081/job/ID/1/console
Dashboard > ID > #1
3359bc3d7a6a: Waiting
4b7c01ed0534: Waiting
5f70bf18a086: Mounted from library/tomcat
4e5b554b7345: Mounted from library/tomcat
bc05267c613b: Mounted from library/tomcat
43c9f8a1dd61: Mounted from library/tomcat
e67e6647dc8d: Pushed
3359bc3d7a6a: Mounted from library/tomcat
f844dcf94898: Mounted from library/tomcat
39cf0ac89a5a: Mounted from library/tomcat
4b7c01ed0534: Mounted from library/tomcat
latest: digest: sha256:2ee5fecd8e741e077de4d3c91a60f20760b5ebc018b3b19b0eeb24f45634638d size: 2409
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
REST API
Jenkins 2.501


rahulsiddharthdh@DHRS: /mn
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Fri Mar 21 04:45:27 UTC 2025
System load: 0.56
Usage of /: 0.8% of 1006.85GB
Memory usage: 14%
Swap usage: 0%
Processes: 39
Users logged in: 0
IPv4 address for eth0: 172.30.157.157
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge
This message is shown once a day. To disable it please create the
/home/rahulsiddharthdh/.hushlogin file.
rahulsiddharthdh@DHRS:/mnt/c/Users/Rahul Siddharth D H$ 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" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: default-storageclass, storage-provisioner
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
rahulsiddharthdh@DHRS:/mnt/c/Users/Rahul Siddharth D H$