

DevOps Day-3

#DOCKER COMMANDS FOR UBUNTU

```
$ sudo apt update -y
```

```
$ sudo apt install docker -y
```

```
$ sudo service docker start (or) sudo systemctl start docker
```

```
$ sudo service docker enable (or) sudo systemctl enable docker
```

DOCKER COMPOSE

Docker Compose is a tool that allows you to define and manage multicontainer Docker applications. It simplifies the process of running multiple containers, their configurations, and their interdependencies. Compose uses a YAML file to define the services, networks, and volumes required for your application.

Docker Compose is a tool which is used to manage multi container-based applications.

Using Docker Compose we can easily setup & deploy multi containerbased applications.

We will give containers information to Docker Compose using YML file (docker-compose.yml)

Docker Compose YML should have all the information related to containers creation.

Docker Compose YML File Looks Like:

```

Windows PowerShell
Pulling db (mysql:latest)...
latest: Pulling from library/mysql
804bb8ae89de: Pull complete
1b515e7ceb69: Pull complete
eaa11c0a9f08: Pull complete
8d18181893b8: Pull complete
e0a910cc8604: Pull complete
bc0c792ca096: Pull complete
8d73d2a73425: Pull complete
4a7e00d87309: Pull complete
27a2553d6a80: Pull complete
69e76254f502: Pull complete
Digest: sha256:9b9d0aab4860798acff13d2a0e3bc26639fe18b83fa5cd3e3d0e16b3ed05dd
Status: Downloaded newer image for mysql:latest
Creating ragul_db_1 ... done
Creating ragul_web_1 ... done
ragul@Admin: $ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS
52d9243ecb73      nginx:latest       "/docker-entrypoint..." 56 seconds ago      Up 55 seconds      0.0.0.0:80->80/tcp, :::80->80/tcp
b1306e37d2a1      mysql:latest       "docker-entrypoint.s..." 56 seconds ago      Up 55 seconds      3306/tcp, 33060/tcp
95c727f860b4      gcr.io/k8s-minikube/kicbase:v0.0.46 "/usr/local/bin/entr..." 25 minutes ago      Up 25 minutes      127.0.0.1:32772->22/tcp, 127.0.0.1:32771->2376/tcp, 127.0.0.1:32770->5080/tcp, 127.0.0.1:32769->8443/tcp, 127.0.0.1:32768->32443/tcp
ragul@Admin: $ sudo docker exec -it ragul_db_1 mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
ragul@Admin: $ sudo docker exec -it ragul_db_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

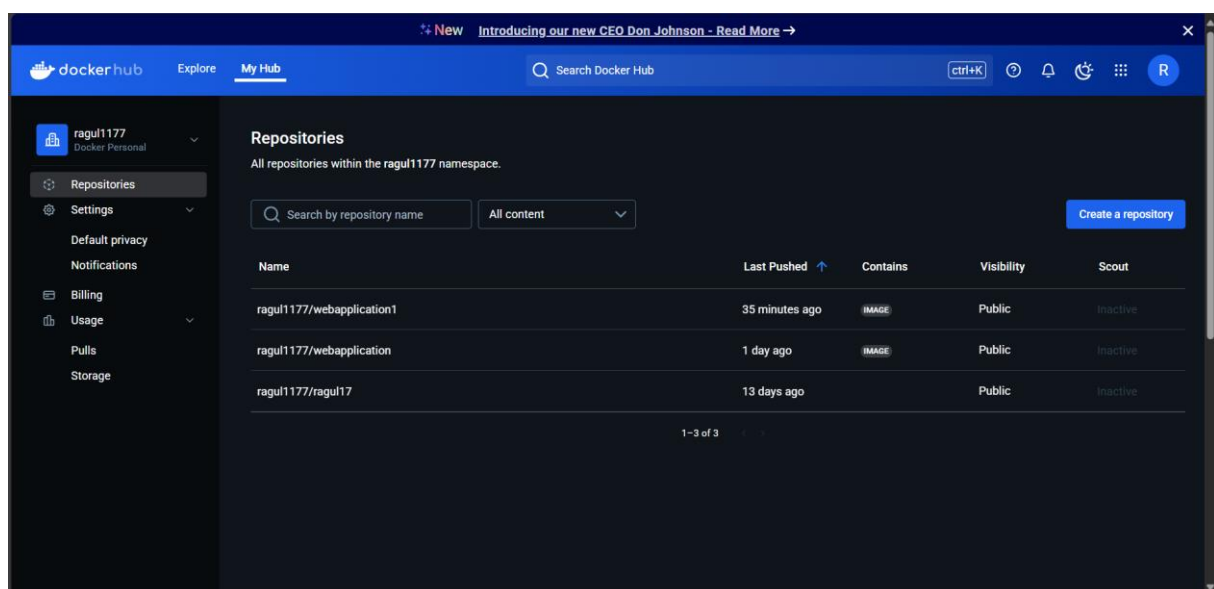
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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>

```



Installed Kubectl using :

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

```
chmod +x kubectl mkdir -p
```

```
~/local/bin mv ./kubectl
```

```
~/local/bin/kubectl
```

and then append (or prepend) ~/local/bin to \$PATH

```
kubectl version --client --output=yaml
```

Minikube Installation:

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

```
minikube start minikube
```

```
status
```

```
kubectl get pod
```

```
kubect get deploy
```

```
kubectl get replica kubectl
```

```
get pod -o wide
```

Dashboard > mvn > #11

- Status
- Changes
- Console Output
- Edit Build Information
- Timings
- Git Build Data
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Thread Dump
- Pause/resume
- Replay
- Pipeline Steps
- Workspaces

Console Output

Download Copy View as plain text

```
Started by user Ragul E S
Obtained Jenkinsfile from git https://github.com/ragules/simple-web-app.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/mvn
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential cred-3
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/mvn/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ragules/simple-web-app.git # timeout=10
Fetching upstream changes from https://github.com/ragules/simple-web-app.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/ragules/simple-web-app.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision cc134d116c29eb1e0edf6cf6e0e089ef646eaf (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
```