Classroom Assignment – Mahesh Bharambe

1. Containers – Running Applications

Q1. Run an Nginx Web Server

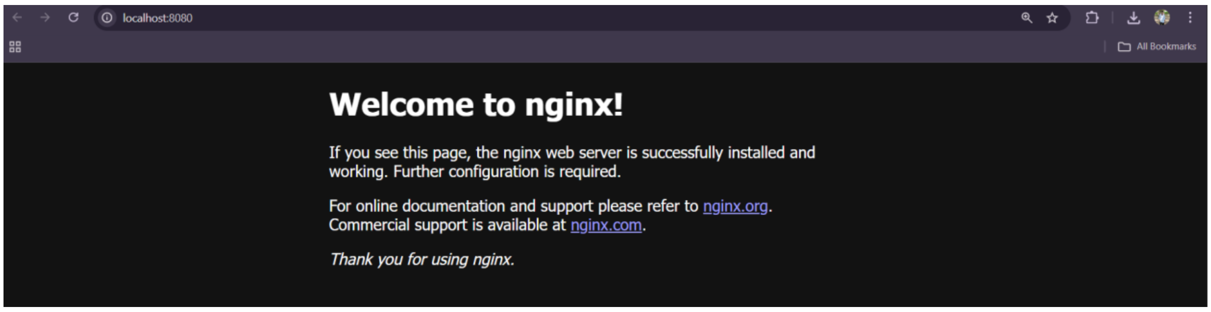
Pull the image from Dockerhub registry .

docker pull nginx:latest

Run the nginx container from nginx:latest

docker run -p 8080:80 --name nginx-cont nginx:latest

Output



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Q2. Run a Python App inside a Container

Pull the image from Dockerhub registry .

docker pull python:3.10-slim

Run the python container from python:3.10-slim

docker run --name python-cont1 --rm python:3.10-slim python -c "print('Mahesh Bharambe')"

Output

Mahesh Bharambe

Q3. Run a MySQL Database

Pull the image from Dockerhub registry .

docker pull mysql:8

Run the mysql container from mysql:8

docker run --name mysql-cont -d -e MYSQL_ROOT_PASSWORD=root mysql:8

Access the container shell

docker exec -it mysql-cont /bin/bash

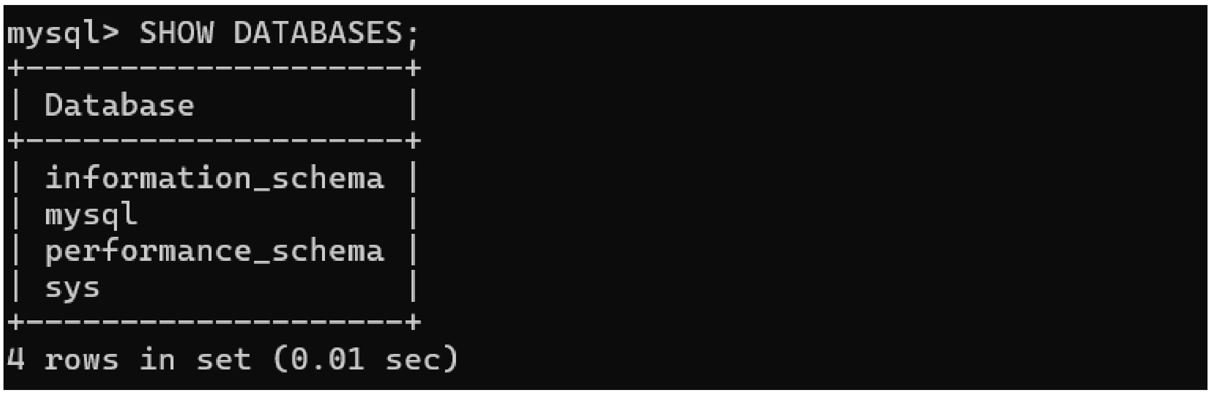
To login in to Mysql

mysql -uroot -p   **(Inside the bash)**

To show Databases

SHOW DATABASES;

Output



```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)
```

----

2. Custom Images – Build Your Own

Q4. Build a Flask Application Image

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/Flask-app

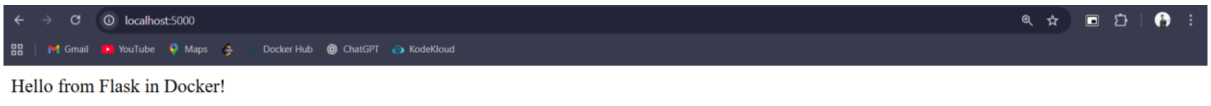
To build to image from dockerfile

docker build -t python-img

To run the container from python image

docker run -d -p 5000:5000 --name flask-app python-img:latest

Output



localhost:5000

Gmail YouTube Maps Docker Hub ChatGPT KodeKloud

Hello from Flask in Docker!

Q5. Create a Custom Nginx Image

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/Nginx-server-docker

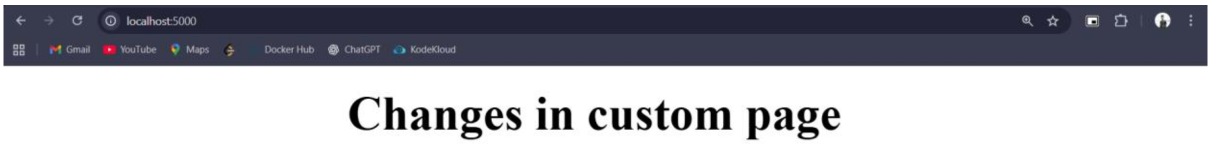
To build to image from dockerfile

docker build -t nginx-html .

To run the container from nginx image

docker run --name nginx-contt -d -p 5000:80 nginx-html

Output



localhost:5000

Gmail YouTube Maps Docker Hub ChatGPT KodeKloud

Changes in custom page

----

3. Volumes – Persisting Data

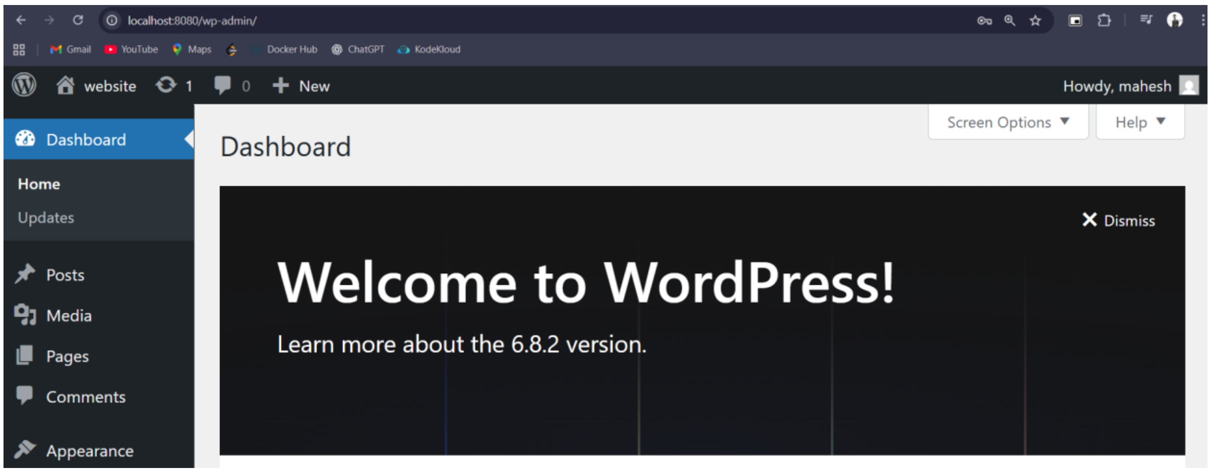
Q6. WordPress with Persistent MySQL

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/wordpress-mysql-docker

To run the containers

docker compose up

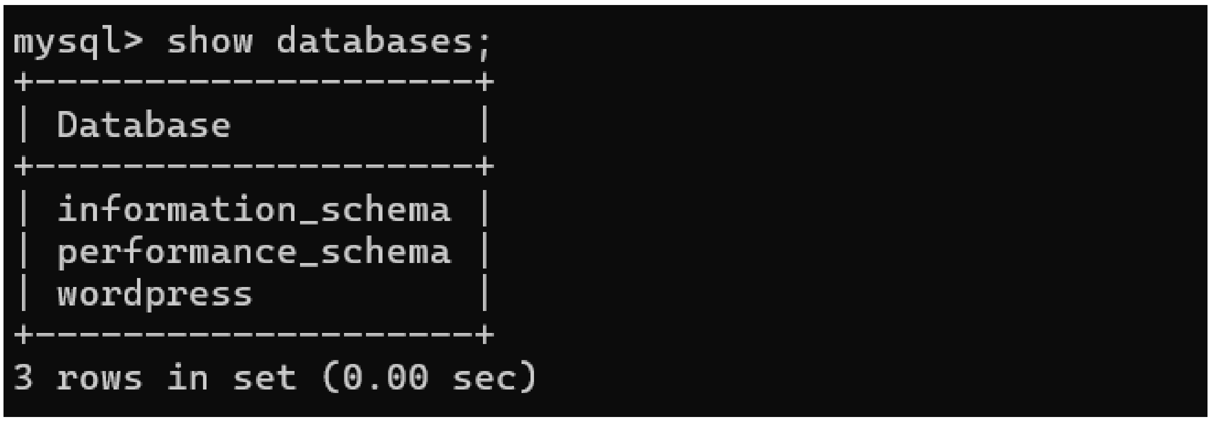
output



Dashboard

Welcome to WordPress!

Learn more about the 6.8.2 version.



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| wordpress          |
+--------------------+
3 rows in set (0.00 sec)
```

Q7. Flask with Bind Mount

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/Flask-app

To build to image from dockerfile

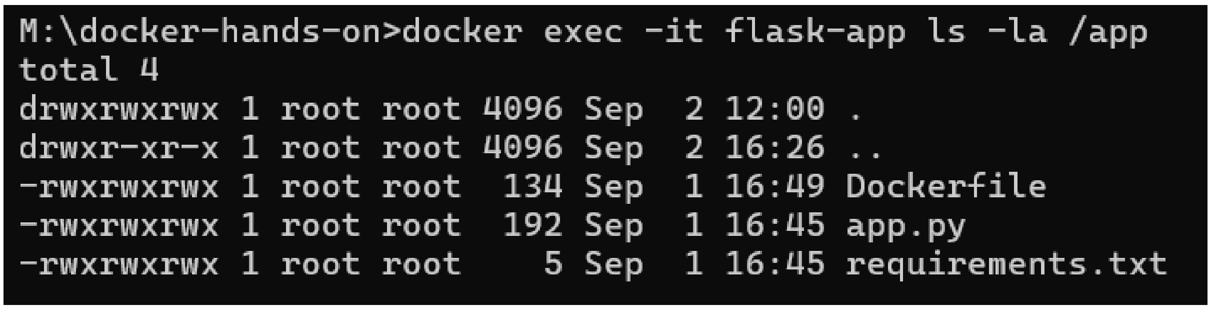
docker build -t python-img .

To run the container from python image

docker run --name flask-app --rm -p 5000:5000 -v M:/docker-hands-on/Flask-app:/app flask-img:latest

To check data is storing or not

docker exec -it flask-app ls -la /app



```
M:\docker-hands-on>docker exec -it flask-app ls -la /app
total 4
drwxrwxrwx 1 root root 4096 Sep  2 12:00 .
drwxr-xr-x 1 root root 4096 Sep  2 16:26 ..
-rwxrwxrwx 1 root root  134 Sep  1 16:49 Dockerfile
-rwxrwxrwx 1 root root  192 Sep  1 16:45 app.py
-rwxrwxrwx 1 root root    5 Sep  1 16:45 requirements.txt
```

---

4. Networks – Multi-Container Applications

Q8. Flask + Redis Counter App

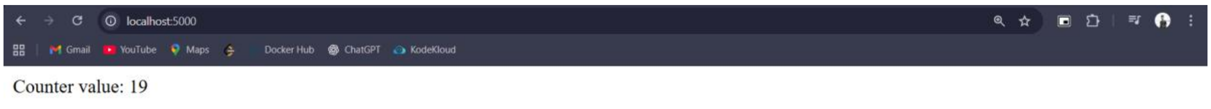
To run the containers of flask and redis

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/Flask-redis-docker

To run the container

docker compose up

Output



Counter value: 19

Q9. MySQL + WordPress Blogging App

To run the containers of mysql and wordpress

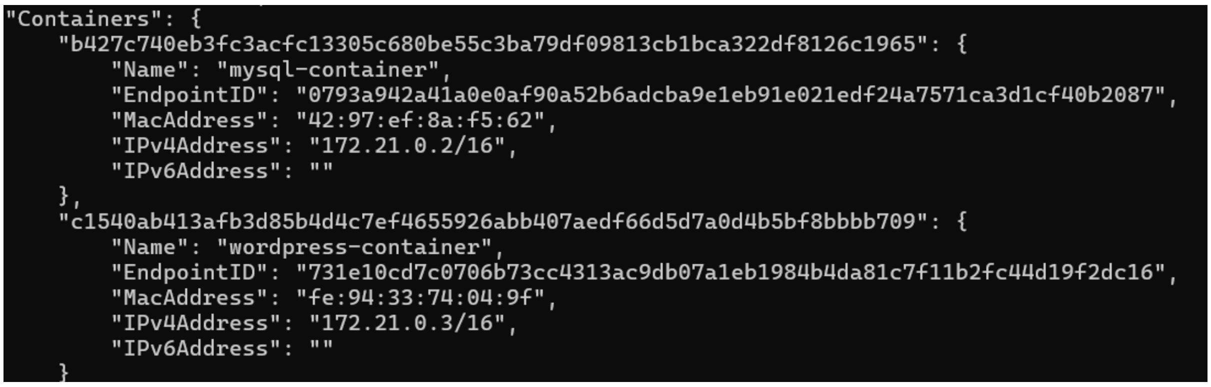
https://github.com/Maheshbharambe45/docker-hands-on/tree/main/wordpress-mysql-docker-network

To run the container

docker compose up

To check both containers in same network

docker network ls wordpress-mysql-docker-network



```
"Containers": {
    "b427c740eb3fc3acfc13305c680be55c3ba79df09813cb1bca322df8126c1965": {
        "Name": "mysql-container",
        "EndpointID": "0793a942a41a0e0af90a52b6adcba9e1eb91e021edf24a7571ca3d1cf40b2087",
        "MacAddress": "42:97:ef:8a:f5:62",
        "IPv4Address": "172.21.0.2/16",
        "IPv6Address": ""
    },
    "c1540ab413afb3d85b4d4c7ef4655926abb407aedf66d5d7a0d4b5bf8bbbb709": {
        "Name": "wordpress-container",
        "EndpointID": "731e10cd7c0706b73cc4313ac9db07a1eb1984b4da81c7f11b2fc44d19f2dc16",
        "MacAddress": "fe:94:33:74:04:9f",
        "IPv4Address": "172.21.0.3/16",
        "IPv6Address": ""
    }
```

Docker Compose

Q10. Convert Flask + Redis App into Docker Compose

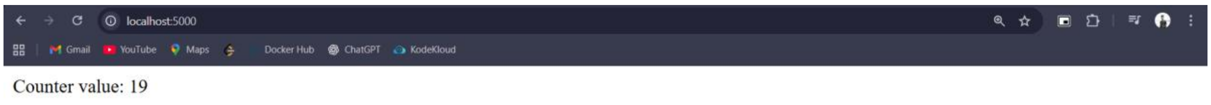
To run the containers of flask and redis

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/Flask-redis-docker

To run the container

docker compose up

Output



Counter value: 19

Q11. Node.js + MongoDB Full Stack Setup

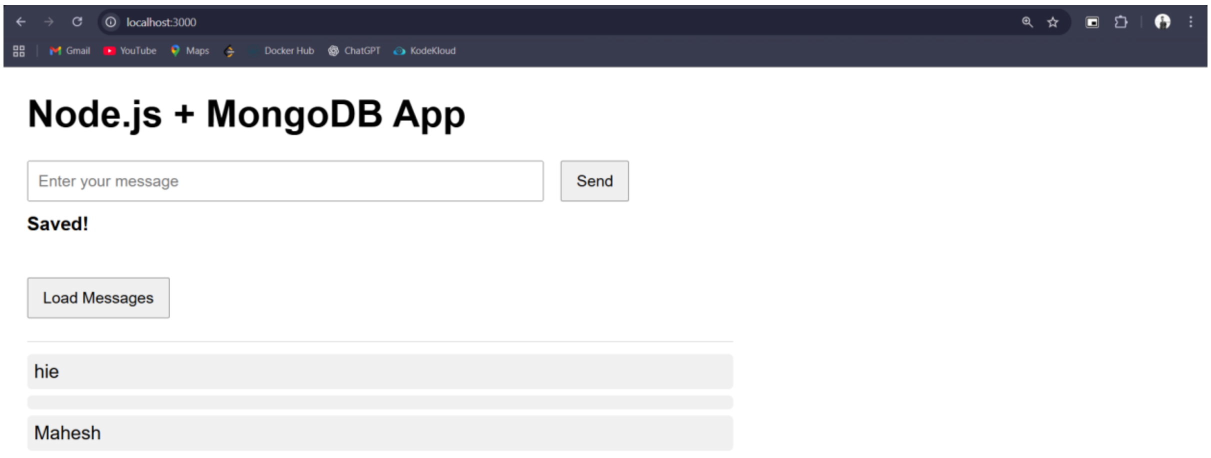
To run the node and MongoDB container

https://github.com/Maheshbharambe45/docker-hands-on/tree/main/Nodejs-mongo-docker

To run the container

docker compose up

Output



Node.js + MongoDB App

Enter your message | Send

Saved!

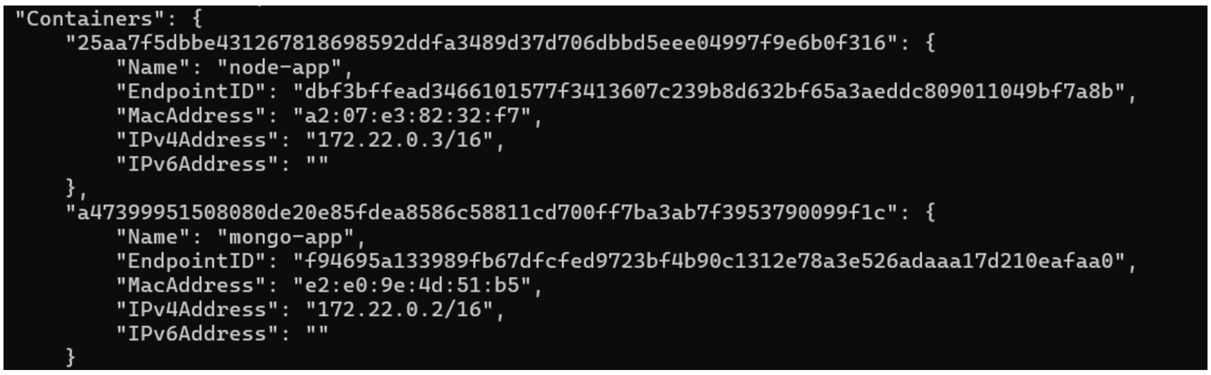
Load Messages

hie

Mahesh

To show the networks

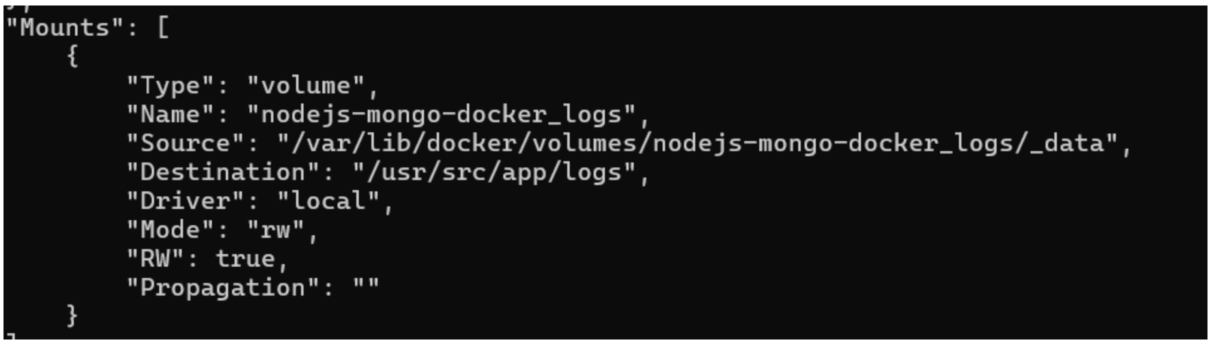
docker network ls



```
"Containers": {
    "25aa7f5dbbe431267818698592ddfa3489d37d706dbbd5eee04997f9e6b0f316": {
        "Name": "node-app",
        "EndpointID": "dbf3bffead3466101577f3413607c239b8d632bf65a3aeddc809011049bf7a8b",
        "MacAddress": "a2:07:e3:82:32:f7",
        "IPv4Address": "172.22.0.3/16",
        "IPv6Address": ""
    },
    "a47399951508080de20e85fdea8586c58811cd700ff7ba3ab7f3953790099f1c": {
        "Name": "mongo-app",
        "EndpointID": "f94695a133989fb67dfcfed9723bf4b90c1312e78a3e526adaaa17d210eafaa0",
        "MacAddress": "e2:e0:9e:4d:51:b5",
        "IPv4Address": "172.22.0.2/16",
        "IPv6Address": ""
    }
```

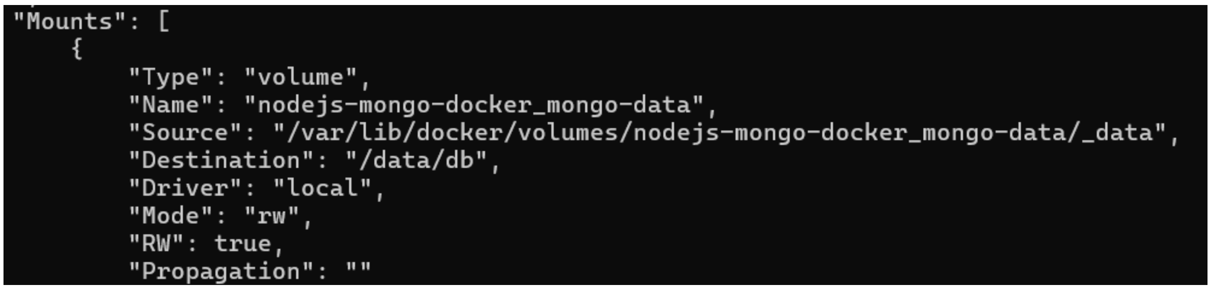
To show the volumes

docker container inspect node-app



```
"Mounts": [
    {
        "Type": "volume",
        "Name": "nodejs-mongo-docker_logs",
        "Source": "/var/lib/docker/volumes/nodejs-mongo-docker_logs/_data",
        "Destination": "/usr/src/app/logs",
        "Driver": "local",
        "Mode": "rw",
        "RW": true,
        "Propagation": ""
    }
```

docker container inspect mongo-app



```
"Mounts": [
    {
        "Type": "volume",
        "Name": "nodejs-mongo-docker_mongo-data",
        "Source": "/var/lib/docker/volumes/nodejs-mongo-docker_mongo-data/_data",
        "Destination": "/data/db",
        "Driver": "local",
        "Mode": "rw",
        "RW": true,
        "Propagation": ""
```
