UNIVERSITY OF PETROLEUM & ENERGY STUDIES

Dehradun

ACO LAB

NAME- YADRISHI DIXIT

BRANCH- COMPUTER SCIENCE ENGINEERING

BATCH- B-4 DEVOPS

SAP ID- 500097959

ROLL NO- R2142211468

SUBMITTED TO- Dr. Hitesh Kumar Sharma

Lab Experiment: Docker Volume

Steps:

Step 1: Create a Docker Volume

Open a terminal on your machine.

Run the following command to create a Docker volume named "my_volume":

```
docker volume create my_volume
```

```
C:\Users\ABC>docker volume create my_volume
my_volume
```

Step 2: Launch Containers with the Volume

Run a container using the volume you created:

```
docker run -it --name container1 -v my_volume:/app/data nginx
```

```
C:\Users\ABC>docker run -it --name container1 -v my_volume:/app/data nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a803e7c4b030: Pull complete
8b625c47d697: Pull complete
4d3239651a63: Pull complete
0f816efa513d: Pull complete
01d159b8db2f: Pull complete
5fb9a81470f3: Pull complete
9b1e1e7164db: Pull complete
Digest: sha256:32da30332506740a2f7c34d5dc70467b7f14ec67d912703568daff790ab3f755
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/09/29 07:01:50 [notice] 1#1: using the "epoll" event method
2023/09/29 07:01:50 [notice] 1#1: nginx/1.25.2
2023/09/29 07:01:50 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/09/29 07:01:50 [notice] 1#1: OS: Linux 5.10.16.3-microsoft-standard-WSL2
2023/09/29 07:01:50 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/09/29 07:01:50 [notice] 1#1: start worker processes
2023/09/29 07:01:50 [notice] 1#1: start worker process 29
2023/09/29 07:01:50 [notice] 1#1: start worker process 30
2023/09/29 07:01:50 [notice] 1#1: start worker process 31
2023/09/29 07:01:50 [notice] 1#1: start worker process 32
```

Enter the container to observe the volume and create a file inside it:

```
touch /app/data/file_in_volume.txt
```

```
C:\Users\ABC>docker exec -it container1 /bin/bash
root@594b269b27f7:/# touch /app/data/file_in_volume.txt
root@594b269b27f7:/# exit
```

```
Exit
```

Run a second container, using the same volume, to verify data persistence:

```
docker run -it --name container2 -v my_volume:/app/data nginx
```

```
C:\Users\ABC>docker run -it --name container2 -v my_volume:/app/data nginx

/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/09/29 07:17:38 [notice] 1#1: using the "epoll" event method
2023/09/29 07:17:38 [notice] 1#1: nginx/1.25.2
2023/09/29 07:17:38 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/09/29 07:17:38 [notice] 1#1: OS: Linux 5.10.16.3-microsoft-standard-WSL2
2023/09/29 07:17:38 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/09/29 07:17:38 [notice] 1#1: start worker processes
2023/09/29 07:17:38 [notice] 1#1: start worker process 29
2023/09/29 07:17:38 [notice] 1#1: start worker process 30
2023/09/29 07:17:38 [notice] 1#1: start worker process 31
```

Enter the second container and check if the file exists:

```
ls /app/data
```

```
Exit
```

```
C:\Users\ABC>docker exec -it container2 /bin/bash
root@7d8de78d869d:/# ls /app/data
file_in_volume.txt
root@7d8de78d869d:/# exit
exit

C:\Users\ABC>
```

Step 3: Cleanup

Stop and remove the containers:

```
docker stop container1 container2
```

```
docker rm container1 container2
```

Remove the volume:

```
docker volume rm my_volume
```

```
C:\Users\ABC>docker volume ls
DRIVER    VOLUME NAME
local     my_volume

C:\Users\ABC>docker stop container1 container2
container1
container2

C:\Users\ABC>docker rm container1 container2
container1
container2

C:\Users\ABC>docker volume rm my_volume
my_volume

C:\Users\ABC>
```

Conclusion:

In this experiment, you learned how to create a Docker volume, associate it with containers, and observed how data persisted between different container instances. Docker volumes are essential for maintaining data integrity, sharing data between containers, and ensuring data persistence even when containers are removed or replaced. This skill is crucial for managing stateful applications and databases within a Dockerized environment.