UPES

UNIVERSITY OF TOMORROW

Application Containerization And Orchestration Lab

Instructor – Dr. Hitesh Kumar Sharma

Submitted By – Swati Pal

SAP ID – 500097368

Enrolment no. – R2142211342

Batch – DevOps B4

Lab Experiment 2: Docker Volume

In this lab experiment, you will learn how to work with Docker volumes, which are used to persist data across containers. Volumes enable data to be stored outside the container filesystem and are crucial for managing data consistency and sharing data between containers.

PREREQUISITES:

Docker installed and running on your machine.

Objective:

Create a Docker volume, use it with a container, and observe how data persists across container instances.

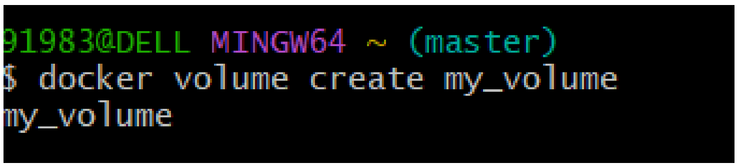
STEPS:

Step 1: Create a Docker Volume

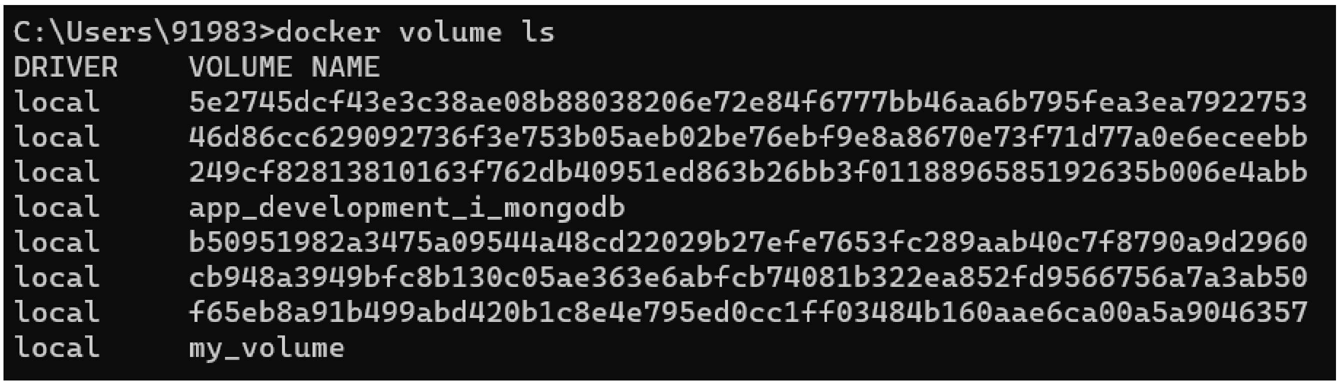
1.a Open a terminal on your machine.

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

```
docker volume create my_volume
```



```
91983@DELL MINGW64 ~ (master)
$ docker volume create my_volume
my_volume
```

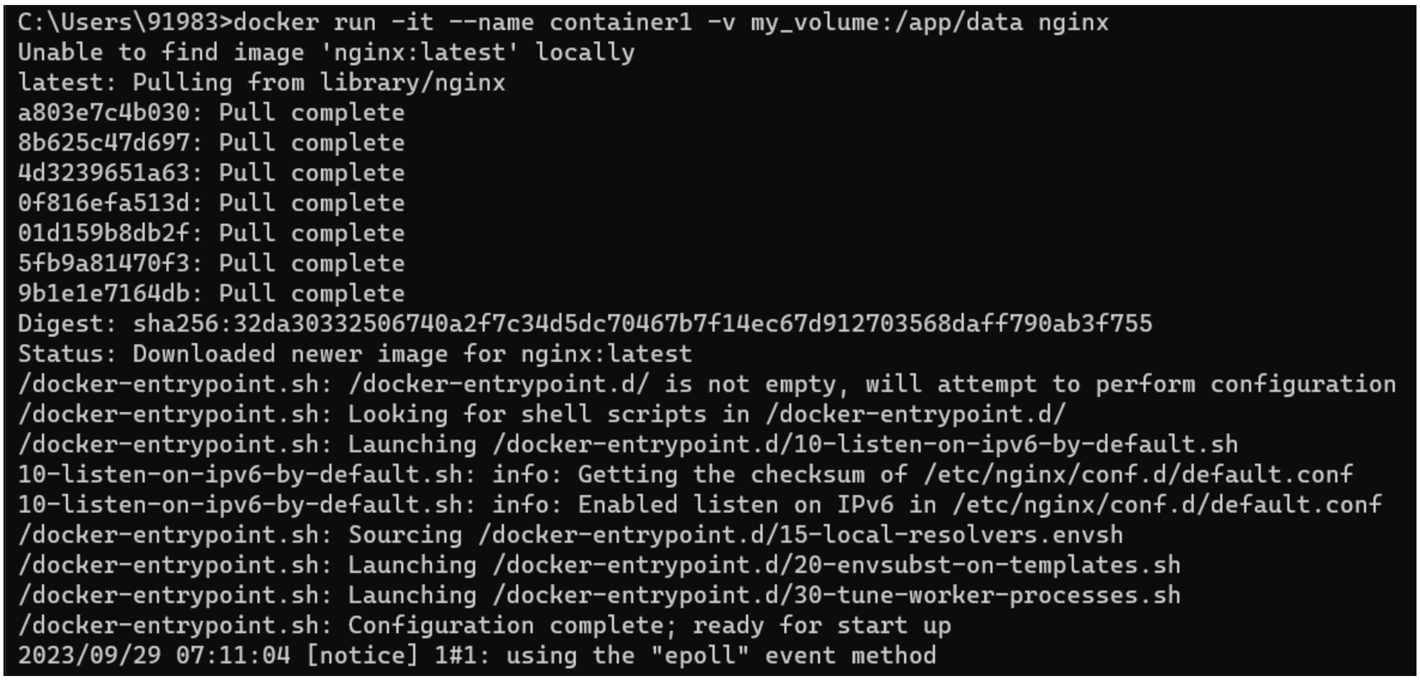


```
C:\Users\91983>docker volume ls
DRIVER    VOLUME NAME
local     5e2745dcf43e3c38ae08b88038206e72e84f6777bb46aa6b795fea3ea7922753
local     46d86cc629092736f3e753b05aeb02be76ebf9e8a8670e73f71d77a0e6eceebb
local     249cf82813810163f762db40951ed863b26bb3f0118896585192635b006e4abb
local     app_development_i_mongodb
local     b50951982a3475a09544a48cd22029b27efe7653fc289aab40c7f8790a9d2960
local     cb948a3949bfc8b130c05ae363e6abfcb74081b322ea852fd9566756a7a3ab50
local     f65eb8a91b499abd420b1c8e4e795ed0cc1ff03484b160aae6ca00a5a9046357
local     my_volume
```

Step 2: Launch Containers with the Volume

2.a Run a container using the volume you created:

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

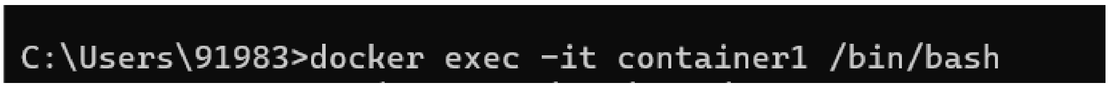


```
C:\Users\91983>docker run -it --name container1 -v my_volume:/app/data nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a803e7c4b030: Pull complete
8b625c47d697: Pull complete
4d3239651a63: Pull complete
0f816efa513d: Pull complete
01d159b8db2f: Pull complete
5fb9a81470f3: 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 07:11:04 [notice] 1#1: using the "epoll" event method
```

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

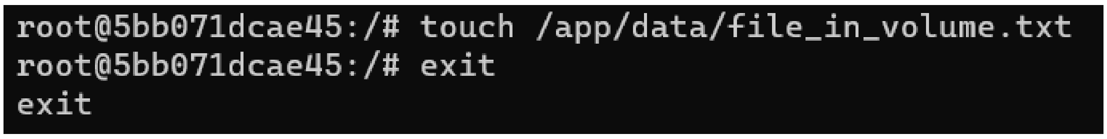
```
touch /app/data/file_in_volume.txt

exit
```

Entering inside the container ‘container1’:



```
C:\Users\91983>docker exec -it container1 /bin/bash
```

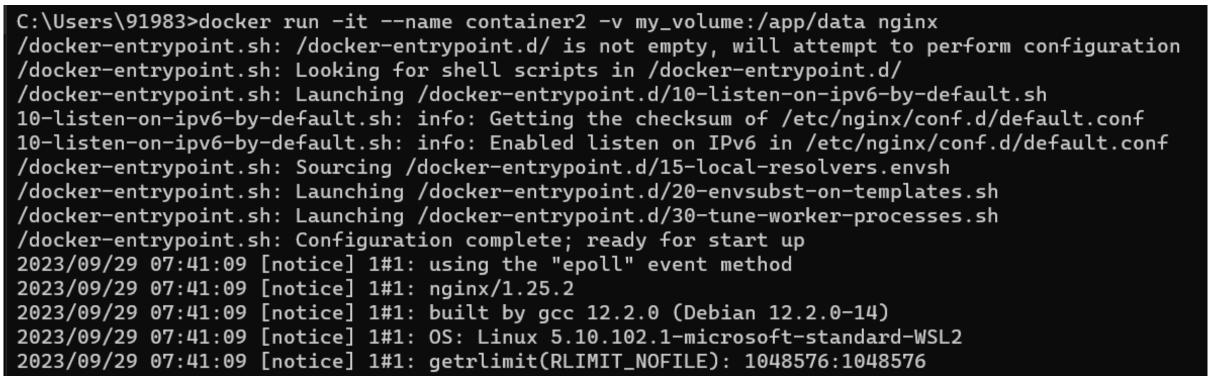
Creating a file inside the ‘container1’:



```
root@5bb071dcae45:/# touch /app/data/file_in_volume.txt
root@5bb071dcae45:/# exit
exit
```

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

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

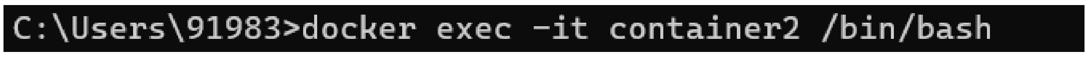


```
C:\Users\91983>docker run -it --name container2 -v my_volume:/app/data nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
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 07:41:09 [notice] 1#1: using the "epoll" event method
2023/09/29 07:41:09 [notice] 1#1: nginx/1.25.2
2023/09/29 07:41:09 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/09/29 07:41:09 [notice] 1#1: OS: Linux 5.10.102.1-microsoft-standard-WSL2
2023/09/29 07:41:09 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
```

2.d Enter the second container and check if the file exists:

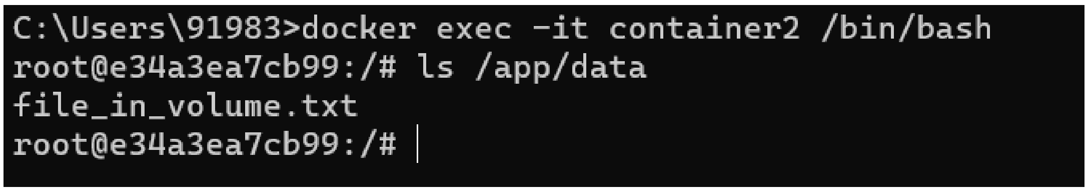
```
ls /app/data

exit
```

Entering inside the container 'container2':



```
C:\Users\91983>docker exec -it container2 /bin/bash
```

Checking if the file 'file_in_volume.txt' exists in the 'container2' or not:



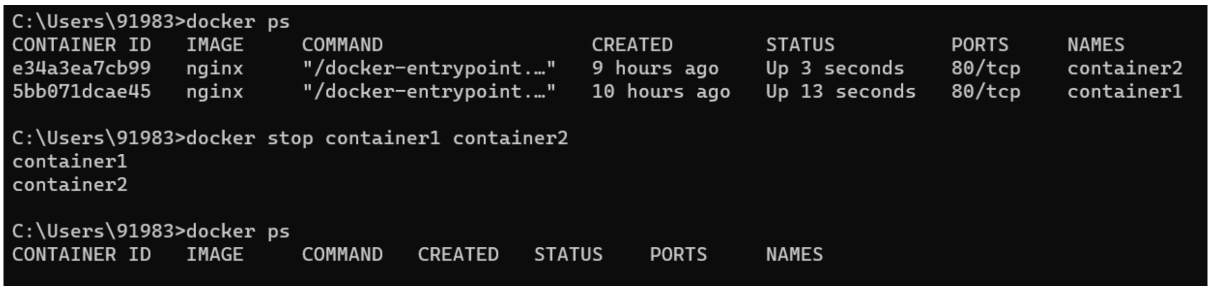
```
C:\Users\91983>docker exec -it container2 /bin/bash
root@e34a3ea7cb99:/# ls /app/data
file_in_volume.txt
root@e34a3ea7cb99:/#
```

Step 3: Cleanup

3.a Stop and remove the containers:

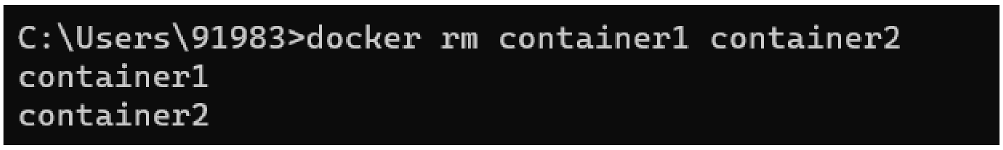
```
docker stop container1 container2

docker rm container1 container2
```

Stopping the containers 'container1' and 'container2':



```
C:\Users\91983>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS          PORTS     NAMES
e34a3ea7cb99   nginx     "/docker-entrypoint.…"   9 hours ago    Up 3 seconds    80/tcp    container2
5bb071dcae45   nginx     "/docker-entrypoint.…"   10 hours ago   Up 13 seconds   80/tcp    container1

C:\Users\91983>docker stop container1 container2
container1
container2

C:\Users\91983>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

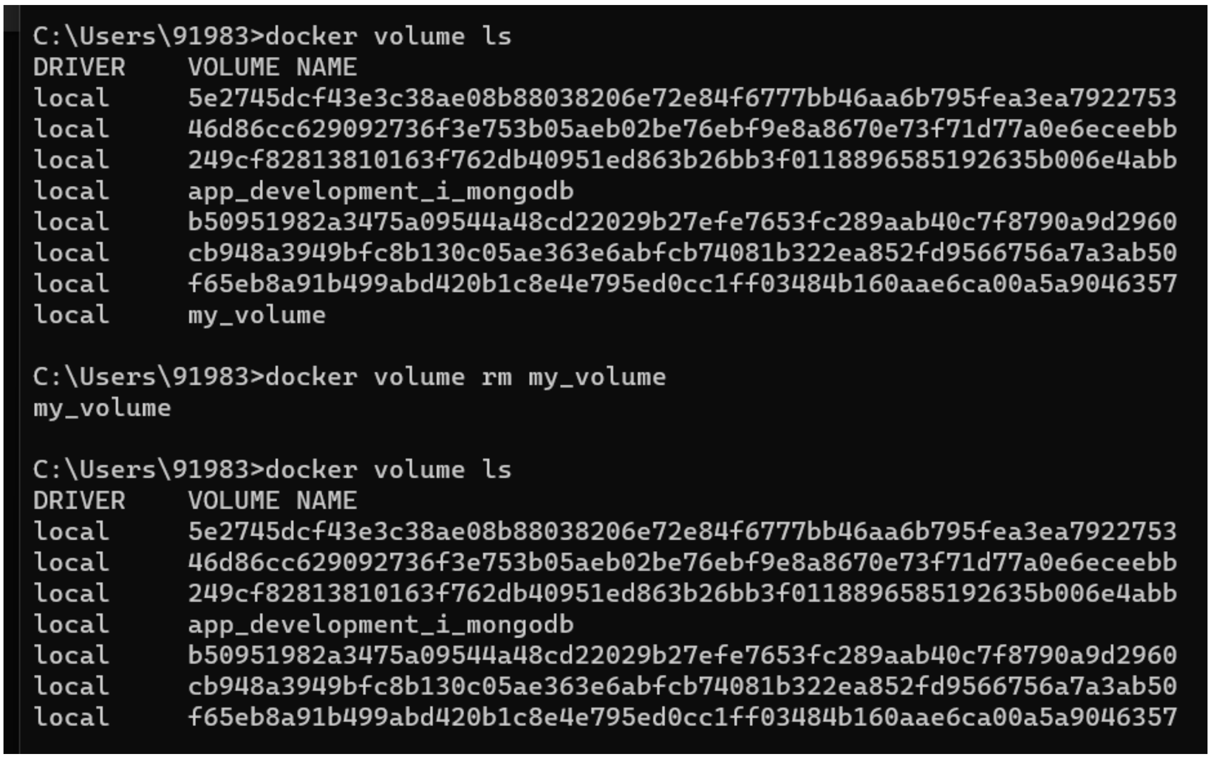
Removing the containers ‘container1’ and ‘container2’:



```
C:\Users\91983>docker rm container1 container2
container1
container2
```

3.b Remove the volume:

```
docker volume rm my_volume
```



```
C:\Users\91983>docker volume ls
DRIVER    VOLUME NAME
local     5e2745dcf43e3c38ae08b88038206e72e84f6777bb46aa6b795fea3ea7922753
local     46d86cc629092736f3e753b05aeb02be76ebf9e8a8670e73f71d77a0e6eceebb
local     249cf82813810163f762db40951ed863b26bb3f0118896585192635b006e4abb
local     app_development_i_mongodb
local     b50951982a3475a09544a48cd22029b27efe7653fc289aab40c7f8790a9d2960
local     cb948a3949bfc8b130c05ae363e6abfcb74081b322ea852fd9566756a7a3ab50
local     f65eb8a91b499abd420b1c8e4e795ed0cc1ff03484b160aae6ca00a5a9046357
local     my_volume

C:\Users\91983>docker volume rm my_volume
my_volume

C:\Users\91983>docker volume ls
DRIVER    VOLUME NAME
local     5e2745dcf43e3c38ae08b88038206e72e84f6777bb46aa6b795fea3ea7922753
local     46d86cc629092736f3e753b05aeb02be76ebf9e8a8670e73f71d77a0e6eceebb
local     249cf82813810163f762db40951ed863b26bb3f0118896585192635b006e4abb
local     app_development_i_mongodb
local     b50951982a3475a09544a48cd22029b27efe7653fc289aab40c7f8790a9d2960
local     cb948a3949bfc8b130c05ae363e6abfcb74081b322ea852fd9566756a7a3ab50
local     f65eb8a91b499abd420b1c8e4e795ed0cc1ff03484b160aae6ca00a5a9046357
```

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.