

Name: Shweta Singh
Sap Id: 500098159
Course & Batch: Btech CSE(DevOps)-B4

Submitted to: Dr. Hitesh Kumar Sharma

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.

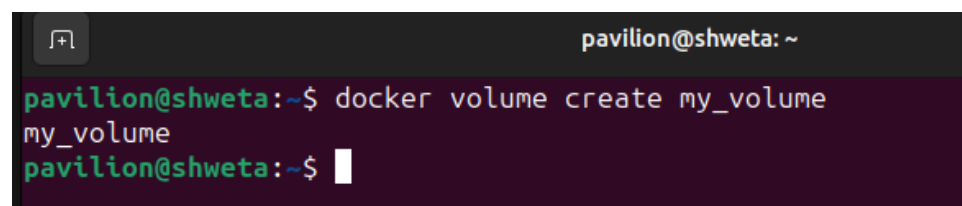
Steps:

Step 1: Create a Docker Volume

Open a terminal on your machine.

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

```
docker volume create my_volume
```

A terminal window with a dark background. The prompt is 'pavilion@shweta: ~'. The command 'docker volume create my_volume' is entered and executed. The output shows 'my_volume' on the next line, followed by a new prompt 'pavilion@shweta:~\$' with a cursor.

```
pavilion@shweta: ~  
pavilion@shweta:~$ docker volume create my_volume  
my_volume  
pavilion@shweta:~$
```

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
```

```
pavilion@shweta:~$ docker run -it --name container1 -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: Configuration complete; ready for start up
2023/09/29 15:59:40 [notice] 1#1: using the "epoll" event method
2023/09/29 15:59:40 [notice] 1#1: nginx/1.25.2
2023/09/29 15:59:40 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/09/29 15:59:40 [notice] 1#1: OS: Linux 6.2.0-33-generic
2023/09/29 15:59:40 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/09/29 15:59:40 [notice] 1#1: start worker processes
2023/09/29 15:59:40 [notice] 1#1: start worker process 29
2023/09/29 15:59:40 [notice] 1#1: start worker process 30
2023/09/29 15:59:40 [notice] 1#1: start worker process 31
2023/09/29 15:59:40 [notice] 1#1: start worker process 32
2023/09/29 15:59:40 [notice] 1#1: start worker process 33
2023/09/29 15:59:40 [notice] 1#1: start worker process 34
2023/09/29 15:59:40 [notice] 1#1: start worker process 35
2023/09/29 15:59:40 [notice] 1#1: start worker process 36
2023/09/29 15:59:46 [notice] 1#1: signal 28 (SIGWINCH) received
2023/09/29 15:59:46 [notice] 36#36: signal 28 (SIGWINCH) received
2023/09/29 15:59:46 [notice] 33#33: signal 28 (SIGWINCH) received
```

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

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

```
pavilion@shweta:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
ce346e7d5086   nginx    "/docker-entrypoint...." 6 minutes ago  Up 6 minutes
80/tcp        container1
pavilion@shweta:~$ docker exec -it ce346e7d5086 /bin/bash
root@ce346e7d5086:/# ls /app/data
root@ce346e7d5086:/# touch /app/data/file_in_volume.txt
root@ce346e7d5086:/# exit
```

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

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

Enter the second container and check if the file exists:

```
ls /app/data
exit
```

```
pavilion@shweta:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
b513bb30bdea   nginx    "/docker-entrypoint...." 16 seconds ago  Up 15 seconds
80/tcp        container2
ce346e7d5086   nginx    "/docker-entrypoint...." 11 minutes ago  Up 11 minutes
80/tcp        container1
pavilion@shweta:~$ docker exec -it b513 /bin/bash
root@b513bb30bdea:/# ls /app/data
file_in_volume.txt
root@b513bb30bdea:/#
```

Step 3: Cleanup

Stop and remove the containers:

```
docker stop container1 container2  
docker rm container1 container2
```

```
pavilion@shweta:~$ docker stop container1 container2  
container1  
container2  
pavilion@shweta:~$
```

```
pavilion@shweta:~$ docker rm container1 container2  
container1  
container2  
pavilion@shweta:~$
```

Remove the volume:

```
docker volume rm my_volume
```

```
pavilion@shweta:~$ docker volume rm my_volume  
my_volume  
pavilion@shweta:~$
```

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.