



Docker

VOLUME

Working with Docker Volumes

Objective

- Create and manage Docker volumes.
- Persist data across container restarts.
- Mount volumes to containers.

Prerequisites

- Docker installed.
 - Basic knowledge of Docker commands and container concepts.
-

Step 1: Create a Docker Volume

Command:

```
docker volume create my_volume
```

```
ubuntu@ip-172-31-83-120:~$ sudo docker volume create my_volume
my_volume
```

Verify Volume:

```
docker volume ls
```

You should see `my_volume` listed.

```
ubuntu@ip-172-31-83-120:~$ sudo docker volume ls
DRIVER      VOLUME NAME
local       my_volume
```

Step 2: Run Container with Volume Mounted

Command:

```
docker run -d --name my_docker_container -v my_volume:/usr/share/nginx/html -p 8012:80 nginx
```

Mounts volume to Nginx's web root.

Verify Container:

```
docker ps
```

```
ubuntu@ip-172-31-83-120:~$ sudo docker run -d --name my_docker_container -v my_volume:/usr/share/nginx/html -p 8012:80 nginx
26844c20c5399de4a7159ae385b735ced9c74b85c7b1a9b9e8e7fba63a9b01a7
ubuntu@ip-172-31-83-120:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
26844c20c539   nginx    "/docker-entrypoint..." 14 seconds ago Up 13 seconds  0.0.0.0:8012->80/tcp, [::]:8012->80/tcp my_docker_container
ubuntu@ip-172-31-83-120:~$ sudo docker exec -it my_docker_container bash
root@26844c20c539:/# echo "<h1>Hello,Docker Volume</h1>">/usr/share/nginx/html/index.html
root@26844c20c539:/# exit
exit
```

Step 3: Interact with the Volume

Enter Container:

```
docker exec -it my_docker_container bash
```

Create HTML File:

```
echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
```

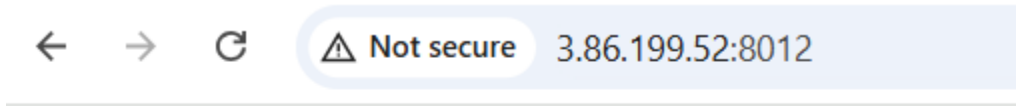
```
exit
```

```
ubuntu@ip-172-31-83-120:~$ sudo docker run -d --name my_docker_container -v my_volume:/usr/share/nginx/html -p 8012:80 nginx
26844c20c5399de4a7159ae385b735ced9c74b85c7b1a9b9e8e7fba63a9b01a7
ubuntu@ip-172-31-83-120:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
26844c20c539   nginx    "/docker-entrypoint..." 14 seconds ago Up 13 seconds  0.0.0.0:8012->80/tcp, [::]:8012->80/tcp my_docker_container
ubuntu@ip-172-31-83-120:~$ sudo docker exec -it my_docker_container bash
root@26844c20c539:/# echo "<h1>Hello,Docker Volume</h1>">/usr/share/nginx/html/index.html
root@26844c20c539:/# exit
exit
```

Test in Browser:

Visit: <http://localhost:8012>

You should see: **Hello, Docker Volume!**



Hello, Docker Volume

Step 4: Test Data Persistence

Stop & Remove Container:

```
docker stop my_docker_container
```

```
docker rm my_docker_container
```

```
ubuntu@ip-172-31-83-120:~$ sudo docker stop my_docker_container
my_docker_container
ubuntu@ip-172-31-83-120:~$ sudo docker rm my_docker_container
my_docker_container
ubuntu@ip-172-31-83-120:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
```

Re-run New Container:

```
docker run -d --name my_second_container -v my_volume:/usr/share/nginx/html -p 8012:80
nginx
```

```
ubuntu@ip-172-31-83-120:~$ sudo docker run -d --name my_second_container -v my_volume:/usr/share/nginx/html -p 8012:80 nginx
71d8d354736613d334f17ea2e46870841eb20e992aeae1ef1327a6a5190caa2d
```

Verify Persistence:

Visit: `http://localhost:8012`

The page should still display: **Hello, Docker Volume!**

← → ↻ ⚠ Not secure 3.86.199.52:8012

Hello,Docker Volume

🧹 Step 5: Clean Up

Stop & Remove Container:

```
docker stop my_second_container
```

```
docker rm my_second_container
```

```
ubuntu@ip-172-31-83-120:~$ sudo docker stop my_second_container
my_second_container
ubuntu@ip-172-31-83-120:~$ sudo docker rm my_second_container
my_second_container
```

Remove Volume:

```
docker volume rm my_volume
```

Verify Removal:

```
docker volume ls
```
