

- Docker Volumes

// A volume is a way to persist data outside of a container's filesystem. When a container is stopped or removed, any data that was stored in its filesystem is lost. However, by creating a volume in Docker, you can store data outside of the container and reuse it across multiple containers, or even persist it after a container is deleted.

// to create a volume in Docker & to check the list of all volumes on your system.

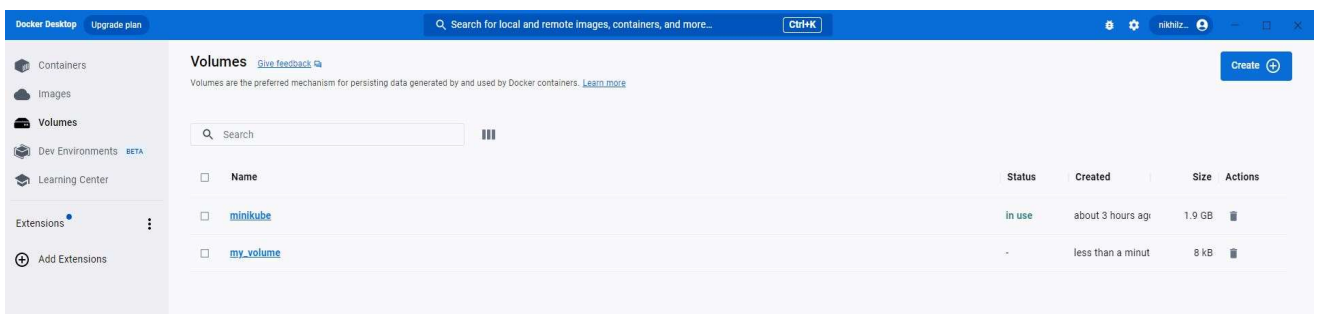


```
C:\Users\lalit> docker volume create --name=my_volume
my_volume

C:\Users\lalit> docker volume ls
DRIVER      VOLUME NAME
local       minikube
local       my_volume

C:\Users\lalit>
```

// The Volume “my\_volume” created on Docker desktop



// inspect the details of a specific volume. It will show you more detailed information about the volume, such as its creation date, labels, and options.



```
C:\Users\lalit> docker volume inspect my_volume
[
  {
    "CreatedAt": "2023-04-27T08:58:10Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/my_volume/_data",
    "Name": "my_volume",
    "Options": {},
    "Scope": "local"
  }
]

C:\Users\lalit>
```

// To mount a volume to a container, we need to use the --mount or -v option when starting the container, followed by the source volume and the target mount point inside the container.  
( docker run --mount source=<volume\_name>,target=<container\_path> <image\_name> )

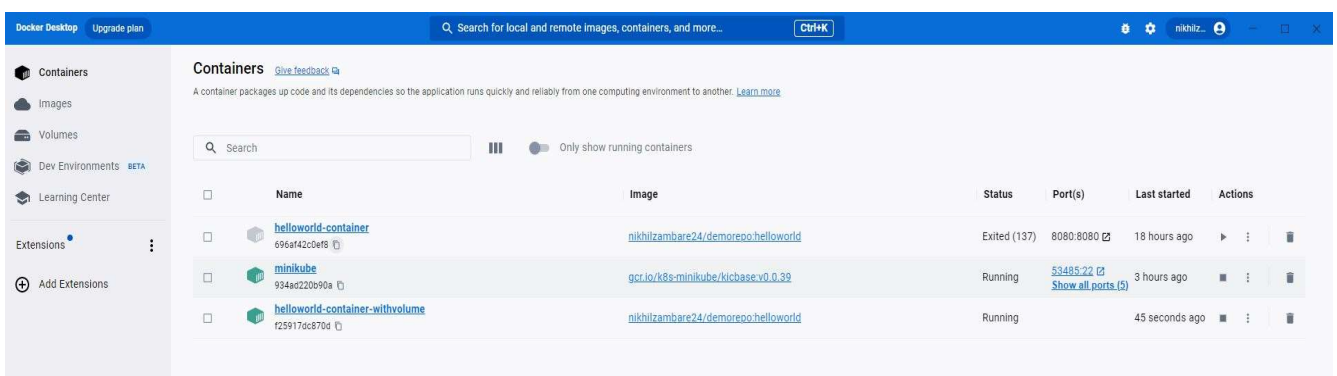
```

C:\Users\lalit> docker run -d --name helloworld-container-withvolume --mount type=volume,source=my_volume,target=/app/data
nikhilzambare24/demorepo:helloworld
f25917dc870dc2649965e1014ce1348da94b11d446fb9359e32e53dcaed152de

C:\Users\lalit> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
NAMES
f25917dc870d   nikhilzambare24/demorepo:helloworld "java -jar /springbo..." 2 minutes ago  Up About a minute  8080/tcp
helloworld-container-withvolume
934ad220b90a   gcr.io/k8s-minikube/kicbase:v0.0.39 "/usr/local/bin/entr..." 3 hours ago    Up 3 hours      127.0.0.1:53485->22/tcp, 127.0.0.1:53486->2376/tcp, 127.0.0.1:53488->5000/tcp, 127.0.0.1:53489->8443/tcp, 127.0.0.1:53487->32443/tcp
cp   minikube

C:\Users\lalit>
```

Here, the source parameter specifies the name of the Docker volume to be mounted to the container, and the target parameter specifies the path inside the container where the volume should be mounted. In this example, we're mounting the volume to the /app/data directory inside the container.



// Created 3 sample volumes (i.e. my\_data, my\_data1 and my\_data2)

```

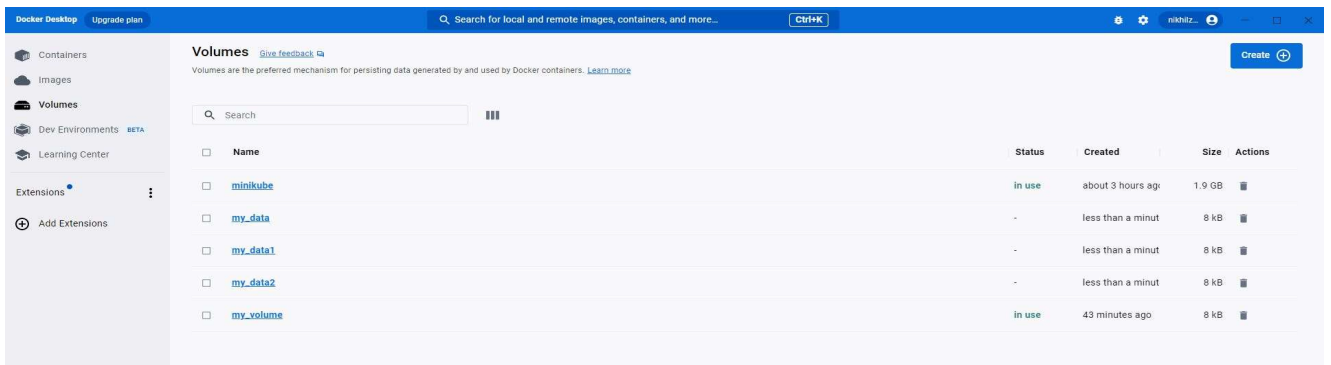
C:\Users\lalit> docker volume create --name=my_data
my_data

C:\Users\lalit> docker volume create --name=my_data1
my_data1

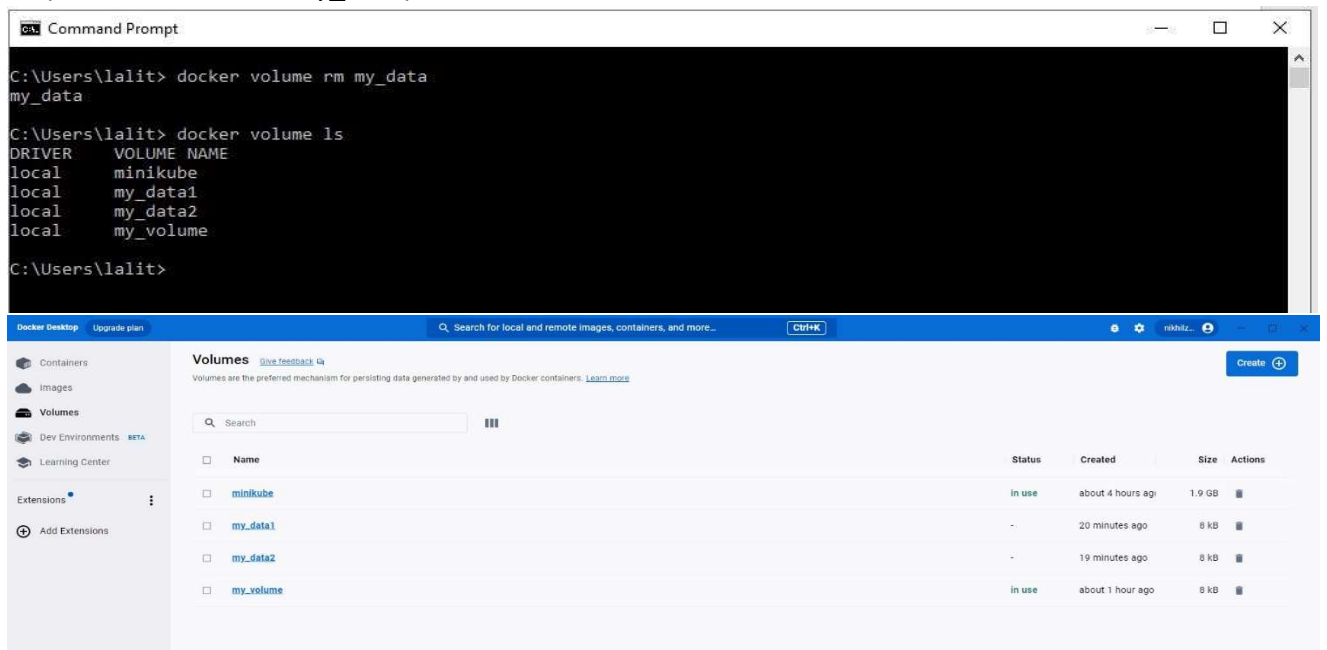
C:\Users\lalit> docker volume create --name=my_data2
my_data2

C:\Users\lalit> docker volume ls
DRIVER      VOLUME NAME
local       minikube
local       my_data
local       my_data1
local       my_data2
local       my_volume

C:\Users\lalit>
```



// docker volume rm volume\_name → Deletes a docker volume  
(it deleted volume my\_data)



// docker volume prune → Deletes all unused Docker volumes  
( it deleted all unused volumes i.e. my\_data1 , my\_data2 )

