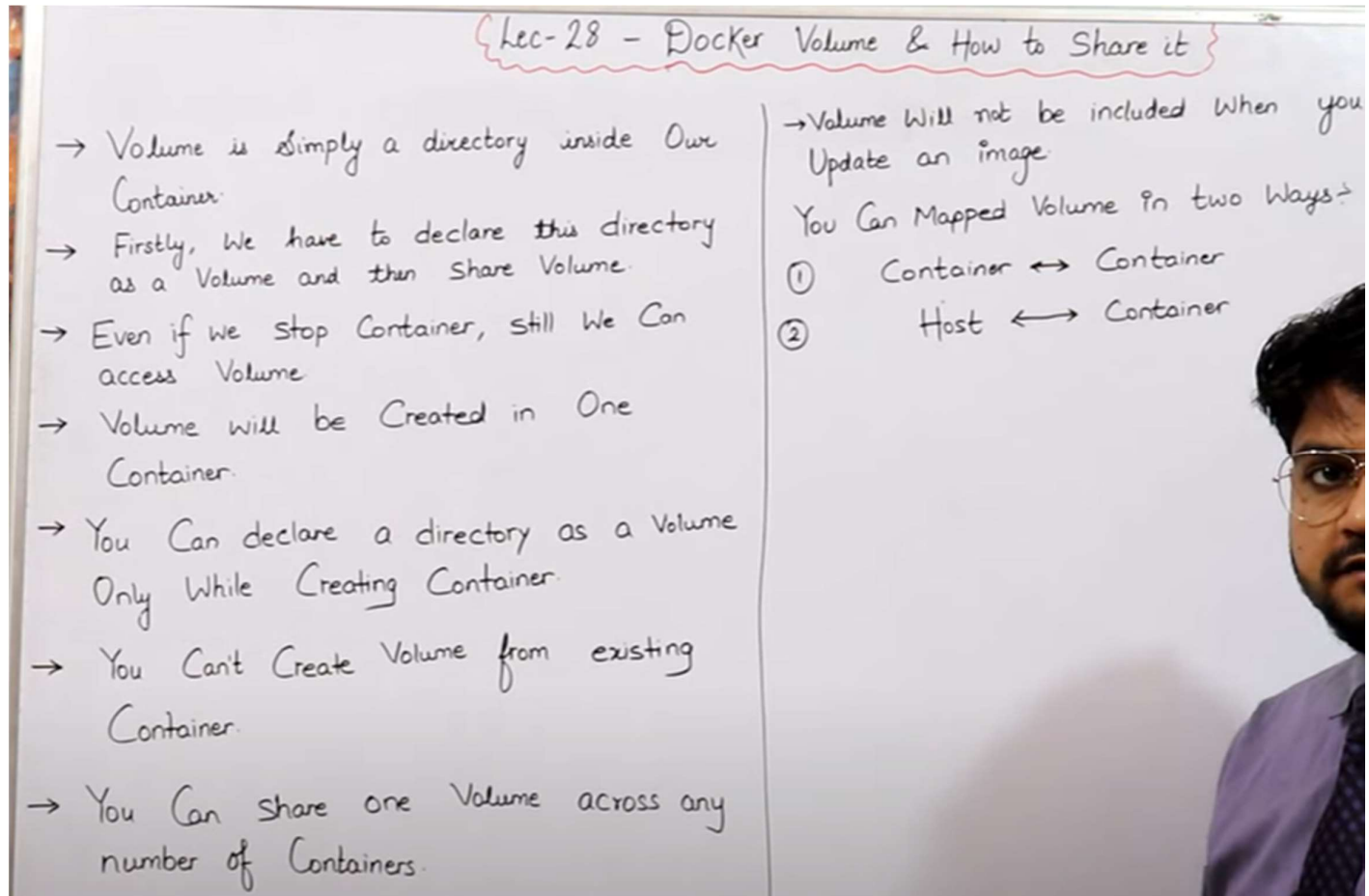


What is Docker Volume | How to Create Volumes | Docker Storage | Docker Volume



- ➔ Hum apne container ke andar kisi bhi ek directory ko volume bana dete hai, Phir vo normal directory nahi rehati vo volume directory ban jati hai.
- ➔ We declare directory as a volume after declare volume we have an option to share volume with anyone.
- ➔ We can share a volume **Container to Container** and **Host to Container**.
- ➔ Hum volume ko hardum container create karne se pehle hi banate hai running container mein hum volume nahi bana sakte.
- ➔ Volume ko hum bahut logo ke sath share kar sakte hai, for example mene volume banaya us volume ko mein multiple user ko de sakta hu, aur user agar volume mein kuch change karta hai toh vo sab ko dikhega jitane logo ke pass volume share hai, but agar volume delete karta hai user toh vo volume keval apne container ke andar se user delete kar payega baki sab ke container mein volume show hoga.

Benefits of Volume

Benefits of Volume

- Decoupling Container from storage
- Share Volume among different Containers
- Attach Volume to Containers
- On deleting Container Volume does not delete.

- ➔ Container kein delete hojane par bhi humara data delete nahi hota jo volume mein data pada hai vo pada rehata hai.
- ➔ Hum apne volume ko multiple account kein sath share kar sakte hai
- ➔ Hum jab docker file likhate hai uske andar hi hum volume ko define kar dete hai.
Running container mein hum volume nahi bana sakte

Create docker volume

Creating Volume from Dockerfile

- ① Create a Dockerfile and Write

```
{ FROM ubuntu  
  VOLUME ["/myvolume1"] }
```

Then Create image from this dockerfile

→ `docker build -t myimage .`

Now Create a Container from this image & Run

`docker run -it --name Container1 myimage /bin/bash`

Now do `ls`, you can see `myvolume1`

Now, Share Volume with another Container

Container1 ↔ Container2

→ `docker run -it --name Container2 (new) --privileged=true --volumes-from Container1 ubuntu /bin/bash`

Now after Creating Container2, `myvolume1` is visible. Whatever you do in One Volume, Can see from other Volume.

→ `touch /myvolume1/samplefile`

→ `docker start Container1`

→ `docker attach Container1`

→ `ls /myvolume1`
you can see `samplefile` here

`exit`

- Hum container banana sein pehale hi volume apane file mein define karate hai aur agar hum apani volume kisi kein sath share bhi karna chahate hai toh bhi container run karane sein pehale hi docker file mein likhana padta hai, ki um kiske sath apani volume share kar rahe hai mera matlab kis container kein sath.
- Lets see agar mene koi volume banayi apane container 1 mein aur jiska naam satyam hai aur vahi volume mene share kiya container 2 aur container 3 kein sath toh unke bhi vaha satyam volume show hoga, aur agar container 2 aur container 3 mein volume mein kuch changes kiye toh vo sabhi container kein volume mein show hoga kyu ki ye common volume rahegi jisko share karoge aap, aur agar unme sein kisi bande mein volume delete kar di toh vo kewal apane container kein andar sein volume delete kar payega baki sab container mein humari volume rahegi.

DOCKER VOLUME CREATE

```
[root@ip-172-31-39-84 ec2-user]# touch satyam1 satyam2 satyam3
[root@ip-172-31-39-84 ec2-user]# ls
satyam1 satyam2 satyam3
[root@ip-172-31-39-84 ec2-user]# vi Dockerfile
```

- Save docker file,
- Yaha mene eak directory ko volume naam sein define kar diya hai, aur ab vo humari volume directory ban gayi.

```
root@ip-172-31-39-84:/home/ec2-user
FROM ubuntu
VOLUME ["/myvolume"]
~
~
```

- Convert docker file to docker image, and we step-2 is making a volume

```
[root@ip-172-31-39-84 ec2-user]# docker build -t myvolumimage .
Sending build context to Docker daemon 8.192kB
Step 1/2 : FROM ubuntu
latest: Pulling from library/ubuntu
7b1a6ab2e44d: Pull complete
Digest: sha256:626ffe58f6e7566e00254b638eb7e0f3b11d4da9675088f4781a50ae288f3322
Status: Downloaded newer image for ubuntu:latest
--> ba6accedd29
Step 2/2 : VOLUME ["/myvolume"]
--> Running in 7f6bfab2df8d
Removing intermediate container 7f6bfab2df8d
--> c68849ce6b04
Successfully built c68849ce6b04
Successfully tagged myvolumimage:latest
[root@ip-172-31-39-84 ec2-user]#
```

- One operating system image & one volume image

```
[root@ip-172-31-39-84 ec2-user]# docker images
REPOSITORY          TAG             IMAGE ID         CREATED          SIZE
myvolumimage        latest         c68849ce6b04    2 minutes ago   72.8MB
ubuntu              latest         ba6accedd29     2 weeks ago     72.8MB
[root@ip-172-31-39-84 ec2-user]#
```

→ Convert image to container

```
[root@ip-172-31-39-84 ec2-user]# docker run -it --name container1 myvolumimage /bin/bash
root@51940a6ee4e0:/# ls
```

→ Now our volume directory is visible.

```
root@51940a6ee4e0:/# ls
bin  dev  home  lib32  libx32  mnt      opt  root  sbin  sys  usr
boot  etc  lib  lib64  media  myvolume  proc  run  srv  tmp  var
root@51940a6ee4e0:/#
```

→ Create a file in volume directory

```
root@51940a6ee4e0:/# cd myvolume/
root@51940a6ee4e0:/myvolume# ls
root@51940a6ee4e0:/myvolume# touch file1 file2 file3
root@51940a6ee4e0:/myvolume# ls
file1 file2 file3
root@51940a6ee4e0:/myvolume# exit
exit
[root@ip-172-31-39-84 ec2-user]#
```

Share container volume to other container

- Mene eak naya container banaya aur us container kein andar mene apane peechale container ki volume ko share kar diya.
- Pehale mene container 3 naam sen eak container banaya then uske baad mene container 1 ki volume ko container 3 kein andar share kar diya.
- **Privileged =true** ka matlab hai ki jiske sath bhi hum apani volume directory share kar rahe usko hum sab permission de rahe ki bhai tum jo chaho changes kar sakte ho volume directory kein andar .

```
[root@ip-172-31-39-84 ec2-user]# docker run -it --name container3 --privileged=true --volumes-from container1 ubuntu /bin/bash
root@774f9dbb1366:/# ls
```

→

```
bin  dev  home  lib32  libx32  mnt      opt  root  sbin  sys  usr
boot  etc  lib  lib64  media  myvolume  proc  run  srv  tmp  var
```



```

root@774f9dbb1366:/# cd myvolume/
root@774f9dbb1366:/myvolume# ls
file1  file2  file3
root@774f9dbb1366:/myvolume#

```

→

→ Now we add some file in our container3, & we see our container1 is also been updated .

→ Agar mene container3 mein koi **file add** kiya hai toh vo file muje container3 mein bhi show hogi jis- jis container mein andar mene vo volume directory share kiya hoga vo sab container mein mujhe **changes** dikhega .

```

root@774f9dbb1366:/# cd myvolume/
root@774f9dbb1366:/myvolume# ls
file1  file2  file3
root@774f9dbb1366:/myvolume# touch hello-india.txt
root@774f9dbb1366:/myvolume# ls
file1  file2  file3  hello-india.txt
root@774f9dbb1366:/myvolume# exit
exit

```

→ Now we go to container1 and check volume directory , hum container3 mein jo kaam kiya hai vo mere container1 mein bhi dikh raha hai .

```

[root@ip-172-31-39-84 ec2-user]# docker attach container1
root@51940a6ee4e0:/# ls
bin  dev  home  lib32  libx32  mnt      opt  root  sbin  sys  usr
boot  etc  lib  lib64  media  myvolume  proc  run  srv  tmp  var
root@51940a6ee4e0:/# cd myvolume/
root@51940a6ee4e0:/myvolume# ls
file1  file2  file3  hello-india.txt
root@51940a6ee4e0:/myvolume#

```

→ If we want to create a volume using command we use this command.

→ Agar hum volume ko bina docker file image mein andar likhe create karna chahate hai toh uske liye hum ye command use karenge iss command mein humara naya container bhi ban jayega aur us container mein andar hum apni volume directory bhi create kar denge , for example agar hum dekhe toh humane satyam-volume2 mein naam mein apni volume directory create kari hai .

→ Command mein **-v** isliye hi likha hai mene ki ki jab mera naya container bane toh usme **satyam-volume-2** naam mein ek volume directory bhi create kar dena.

```

[root@ip-172-31-39-84 ec2-user]# docker run -it --name container4 -v /satyam-volume2 ubuntu /bin/bash
root@a8a844b0ce65:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  sbin  sys  usr
boot  etc  lib  lib64  media  opt  root  satyam-volume2  srv  tmp  var
root@a8a844b0ce65:/#

```

Volumes (Host - Container)

- Verify files in /home/ec2-user
 - `docker run -it --name hostCont -v /home/ec2-user:/rajput --privileged=true ubuntu /bin/bash`
 - `cd /rajput`
Do `ls`, now you can see all files of host machine
 - `touch rajputfile` (in Container)
`exit`
- Now check in EC2 machine, you can see this file

Some other Commands

- `docker volume ls`
- `docker volume create <volumename>`
- `docker volume rm <volume-name>`
- `docker volume prune`
{ It removed all unused docker }
Volume
- `docker volume inspect <volumename>`
- `docker container inspect <ContainerName>`

➤ docker volume ls

humare jitane bhi volume hai sab show ho jayegi

➤ docker volume prune

humari koi bhi volume jo kahi use nahi ho rahi hai usko yein delete kar deta hai

➤ docker volume inspect <volume name>

humare volume mein kya-kya juda hai vo puri detail miljayegi is command mein.

➤ docker container inspect <container name>

hum agar apne container mein judi cheeze janana chahate hai toh ye use karenge .

Mapping

→ Mapping ka kaam hota hai ki jo bhi humare user ke andar file padi hai us file ko container ke andar volume mein daal do aur jab bhi container ke volume mein kuch changes hogi toh vo humare user ke andar bhi dikhegi.

→ Agar mere kisi user ke andar file padi hai aur us file ko mein apne new container ke andar new volume define karke uske andar le jana chah raha toh mein ye karunga.

➤ Humare ec2- user ke andar kuch file padi hai ab is file ko mein chahata hi new container bana ke uske andar volume banau aur ye file us volume mein bhi copy hojaye

```
[root@ip-172-31-39-84 ec2-user]# ls
Dockerfile satyam1 satyam2 satyam3
[root@ip-172-31-39-84 ec2-user]# cd ..
[root@ip-172-31-39-84 home]# ls
ec2-user
[root@ip-172-31-39-84 home]# cd ec2-user/
```

➤ hostname mere new container ka naam hai, ec2-user apne user ka naam diya hai ki is user ke andar jo file padi hai us file ke utha ke new volume create kar do ek new container ke andar

```
[root@ip-172-31-39-84 ec2-user]# docker run -it --name hostname -v /home/ec2-user:/NewVolume --privileged=true ubuntu /bin/bash
root@3f5381356690:/# ls
NewVolume boot etc lib lib64 media opt root sbin sys usr
bin dev home lib32 libx32 mnt proc run srv tmp var
```

➤ Hum dekh sakte hai humara volume create ho gaya hai aur us volume ke andar hum jaake dekhenge toh vo sab file aagayi jo humare user ke andar thi.

```
root@3f5381356690:/# cd NewVolume/
root@3f5381356690:/NewVolume# ls
Dockerfile satyam1 satyam2 satyam3
root@3f5381356690:/NewVolume#
```

→ Abhi mere user ke andar keval itani hi file hai

```
[root@ip-172-31-39-84 ec2-user]# ls
Dockerfile satyam1 satyam2 satyam3
```

→ Agar mene apne container ke volume ke andar new file banayi hai toh vo mere user ke andar bhi dikhega.

```
root@3f5381356690:/NewVolume# ls
Dockerfile satyam1 satyam2 satyam3
root@3f5381356690:/NewVolume# touch file1 file2
root@3f5381356690:/NewVolume# ls
Dockerfile file1 file2 satyam1 satyam2 satyam3
root@3f5381356690:/NewVolume# exit
exit
```

→ Humare user ke andar file1 aur file2 dikhene lagi

```
[root@ip-172-31-39-84 ec2-user]# ls
Dockerfile file1 file2 satyam1 satyam2 satyam3
[root@ip-172-31-39-84 ec2-user]#
```