Volume Use Case

Point 1: Create a Docker File like

```
bhushan@ubuntu:~/Docker-Pract$ vi Dockerfile
bhushan@ubuntu:~/Docker-Pract$ cat Dockerfile
FROM ubuntu
VOLUME ["/myvol"]
```

Point 2: Create bpt_image from the above Dockerfile

```
bhushan@ubuntu:~/Docker-Pract$ docker build -t bpt_image .
Sending build context to Docker daemon  6.144kB
Step 1/2 : FROM ubuntu
 ---> 08d22c0ceb15
Step 2/2 : VOLUME ["/myvol"]
 ---> Running in 54d9a479a4a9
Removing intermediate container 54d9a479a4a9
 ---> 8060dc4d7b4b
Successfully built 8060dc4d7b4b
Successfully tagged bpt_image:latest
bhushan@ubuntu:~/Docker-Pract$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
bpt_image       latest    8060dc4d7b4b   46 seconds ago   77.8MB
cust_nginx      v3        43655163b282   14 hours ago     237MB
cust_nginx      v2        24f44ea06409   14 hours ago     237MB
cust_nginx      v1        d1db1b99bccc   14 hours ago     237MB
bpt_image       v1        22010236136a   15 hours ago     77.8MB
pratik_image    latest    c4f518be9c8d   24 hours ago     231MB
somnath_image   latest    8b81933ccb1d   24 hours ago     77.8MB
nginx           latest    080ed0ed8312   43 hours ago     142MB
ubuntu          latest    08d22c0ceb15   3 weeks ago      77.8MB
```

Point 3: Now Create container having name old_container from the bpt_image. After creating, check whether myvol is present in the directory structure or not. If present go inside the myvol directory. Create 4 files inside the myvol directory.

```
bhushan@ubuntu:~/Docker-Pract$ docker run -it --name old_container bpt_image:latest /bin/bash
root@6bcca08a17b2:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  myvol  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@6bcca08a17b2:/# cd myvol/
root@6bcca08a17b2:/myvol# touch {1..4}.txt
root@6bcca08a17b2:/myvol# ls
1.txt  2.txt  3.txt  4.txt
root@6bcca08a17b2:/myvol# exit
exit
```

Point 4: Now, create another container having name new_container and specify privileged is true and share the old container volume to this new container for that use –volumes-from argument. After creation of new_container check whether myvol directory is reflect or not. Go inside the myvol and check whether all files are present or not.

```
bhushan@ubuntu:~/Docker-Pract$ docker run -it --name new_container --privileged=true --volumes-from old_container ubuntu /bin/bash
root@875cbb9600de:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  myvol  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@875cbb9600de:/# cd myvol/
root@875cbb9600de:/myvol# ls
1.txt  2.txt  3.txt  4.txt
root@875cbb9600de:/myvol# exit
exit
```

Point 5: Now, start the old container and go inside that container. Check myvol directory. Go inside the myvol and create one 1.html here.

```
bhushan@ubuntu:~/Docker-Pract$ docker start old_container
old_container
bhushan@ubuntu:~/Docker-Pract$ docker attach old_container
root@6bcca08a17b2:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  myvol  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@6bcca08a17b2:/# cd myvol/
root@6bcca08a17b2:/myvol# ls
1.txt  2.txt  3.txt  4.txt
root@6bcca08a17b2:/myvol# touch 1.html
root@6bcca08a17b2:/myvol# exit
exit
```

Point 6: Now, start the new container and go inside that container. Check myvol directory. Go inside the myvol and check whether new created file 1.html is reflect here or not. If it is reflected then it means that we can share directory from container to container by using VOLUME. If we change inside the directory it will reflected inside the container.

```
bhushan@ubuntu:~/Docker-Pract$ docker start new_container
new_container
bhushan@ubuntu:~/Docker-Pract$ docker attach new_container
root@875cbb9600de:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  myvol  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@875cbb9600de:/# cd myvol/
root@875cbb9600de:/myvol# ls
1.html  1.txt  2.txt  3.txt  4.txt
root@875cbb9600de:/myvol#
```

Note:- All the changes are reflected vice versa also.