Docker Volume

Containers are very small images and are very light weight and main focus is to start and stop quickly and frequently. This is the main reason usually important persistent data is not stored inside the containers.

If we have to store data pin persistent format then we use volumes. Using volumes data can be shared between containers as well.

Old method of volume creation

docker run -v /hostdir:/var/lib/containerdir -d mysql

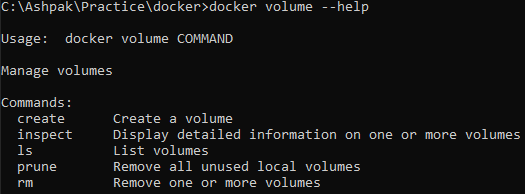
instead of -v we can use –volume as well . -d option here is just for detached mode.

New method of volume creation

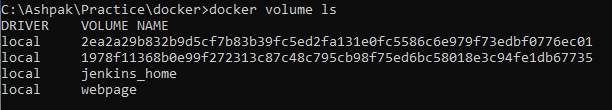
New method use -mount switch instead of -v to create mount volume

## Manage volumes

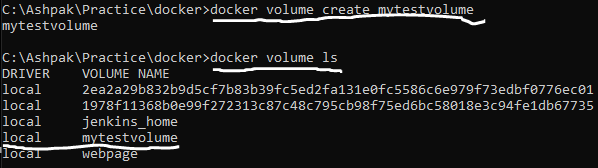
Docker --volume is used to manage volumes created on host machine by containers. It supports basic operations as shown in help manual below for create, inspect, list, prune and remove



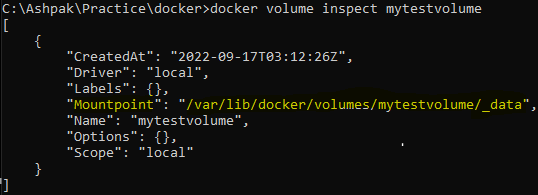
docker volume ls is for listing all existing volumes on host machine



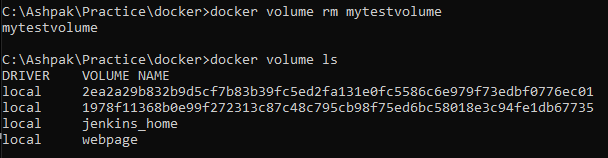
docker volume create <<volumename>> is used for creating a new volume



docker volume inspect <<volumename>> is used for getting details of volume. We see where volume is stored on local disk in Mountpoint key



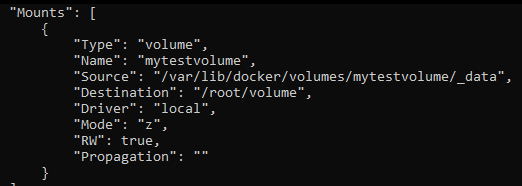
docker volume rm <<volumename>> is used for deleting volume



Mount our created volume to container using –mount switch. Note in mount params there is NO space in ‘source’,’destination’ values. Source and destination can also be referred as src, dst or source, target.

docker run --name my\_nginx\_with\_volume --mount source=mytestvolume,destination=/root/volume -d nginx

if we run docker inspect <<ContainerID>> command then in ‘Mounts’ section we get follow details which also confirms read write as true which shows by default containers can read-write to volumes



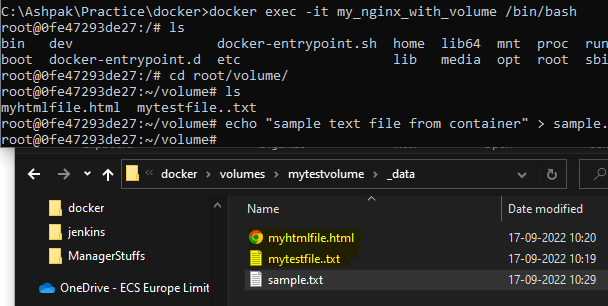
Since volumes stay outside container on a host machine so even if container is stopped, deleted, restarted all the data stored on volume stays intact until we delete the volume manually.

On windows machines if we are running docker using docker desktop then volume scan be found here :

**\\wsl$\docker-desktop-data\version-pack-data\community\docker\volumes**

Now if we manually create following yellow highlighted files in windows explorer on host machine

Then we can see them from inside container as well and we can create files in container and see them in host machine as well.



We can attach single volume from host machine to multiple containers and can also control its success level. Ex. now let’s attach same volume to one more container in read only mode.



\*\* same volume can be attached to multiple containers and we can control using readonly flag is any individual container needs only read access.

## Temp volumes

All volumes we have seen till now are persistent volumes and when container is stopped, deleted they stays on disk and have their data intact, but what if we want a volume which is using host disk but not persistent and it is available only till docket container is there?

We can use temp volumes. Temp volumes created with type=tmpfs param to –mount switch and note that while creating temp volume we don’t have to provide ‘src’ option because we don’t need it to per persistent on specific location. Temp volumes also can not be shared between multiple containers.



If we inspect docker container then we see temp volumes are attached of type tmpfs

