.VOLUME

Volume are preferred mechanism for persisting data generated or used by containers.

Bind mount are depended on the directory structure of host machine where as volumes are completely managed by the Docker

Volume advantage over bind mount

They are easier to backup or migrate than bind mounts

Can be manged using docker cli or api

Works same on linux as well as windows containers

Can be safely shared among multiple conatianers.

Volumes driver allows us to to store volume any where on host or on cloud.

Their content can be pre-populated by containers.

Volume are better choice for persistent data storage as volume doenot increase the container size and it contains exist outside containers life cycle.

IF containers are using non persitant data it is better to use tmpf mount.

This instruction creates a mount point with specified name and mark it as holding externally mounted volumes from native host and other containers the value can be a plain text or Json array

Traditionally –volume command was used for stand alone container and –mount was used for swarms container

But after Docker 17.6 –mount can be used for both standalone as well as for swarm containers

-v or –volume containers 3 fields separated by : the field must be in specified order.

1) name of the volume it is unique for a given host name for anoynomus volume this field is omitted

2) Path where file or directory need to be mounted

3) Optional comma separated list of options like ro etc

--mount contains key value pair separated by commas, This is more verbose and order of key value pair is not important

1) Type of volume can be a volume , bindmout or tmpfs

2) source it is usually the name of the voume , for anoynomus volume this field is omitted.

2) dst / destination its path where the file or directory should be mounted

3) readonly this causes the mount to be read only.

Example

docker service create –mount ‘type=volume,src=<volume name >,dst=<volume path>, volue-driver=local –name my app

Volumes can be created and managed outside the scope of the container

docker volume create my\_volume : to create a new volume

docker volume ls : to list all the volume present

docker volume inspect : To sea the meta data of volume

docker volume rm my\_vol : to remove a existing volume

Note : if you start a container with a volume that doesnot exist docker creats a volume for you

HandsOn task

docker container run –d –p 80:80 –mount src=my\_vol,dst=/app nginx ---name devtest

docker volume inspect devtest

docker container stop

docker image rm devtest

docker volume rm my\_vol1

If you use volume-driver = local non of the container will be able to share a volume.

Example

docker service create –replica=5 –name devtest-service –mount src=<volumename> /

dest=<volume path> ngnix:latest

When you create a volume at run tyoe by usding –volume or –mount option the directory or file contain are copied into mentioned directories.

Example

docker conatainer run –d –p 80:80 –nam proxy –mount src=<name of volume> \

,target=/usr/share/nginix/latest nginx:latest

Note multiple container can be mounted to same volume and can have read-write or read only privileges.

Example

docker container run –d –p 80:80 --name proxy –mountsrc=nginx\_vol, \

target-/user/share/nginx/html, readonly

use docker volume inspect command to check readonly privilege for the volume ‘

Volume driver

They are mainly used to incorporate fault tolerance in your system. They aca be used to wirte dat to NFS amazon S3 or any other cloud system.

They allow you to abstract underlying storage system from the application logic.

Volume driver option can be used when you create volume with volume create command or when you create volume a container run time . use vieux/sshfs volume drivers.

I ntial setup of volume driver to install plugin for vieux or sshfs driver

docker plugin install –grant-all-permission vieux/sshfs

docker volume create --driver vieux/sshfs –o sshcmd=test@node2:/home/test –o password <your password> sshvolume

docker container run –d –name my\_cont –volume-driver vieux/sshfs \

--mount src=my\_vol,target=/app,volume-opt=sshcmd=test@node2:/home/test,volume-opt=password=mypassword nginx:lates

There are two types of volume named volume have specified source from outside container and anyoumous volume : have no specified source you can instruct docker engine or daemon to delete this volume when container is deleted.

Example

Docker container run –rm –v /foo -v myvol:bar:ro nginx

When container is removed the name volume for myvol:/bar is retained where as /foo is deleted

To remove all the unused volume use commad

docker volume prune