**Docker volumes**

<https://docs.docker.com/storage/volumes/>

Container data does not persist.

When stop the container all the data get lost.

Can persist container data to the host machine.

There are 3 ways of doing that.

1. Host machines var/lib/docker/volumes folder - volumes (sudo ls var/lib/docker/volumes)
2. Anywhere in the host machine’s file system - bind mounts
3. Host machine’s memory - tmpfs

Use to avoid persistence of state or sensitive data

**Create & manage volumes**

**(Volumes can be created outside the scope of any container)**

1.

Create volume

sarala@sarala-pc:/$ docker volume create my-vol-1

my-vol-1

This volume is created with in the host

2.

List down the volumes

sarala@sarala-pc:/$ docker volume ls

DRIVER VOLUME NAME

local my-vol-1

3.

Inspect a volume

sarala@sarala-pc:/$ docker volume inspect my-vol-1

[

{

"CreatedAt": "2019-01-06T06:17:53Z",

"Driver": "local",

"Labels": {},

"Mountpoint": "/mnt/sda1/var/lib/docker/volumes/my-vol-1/\_data",

"Name": "my-vol-1",

"Options": {},

"Scope": "local"

}

]

4.

Remove a volume

sarala@sarala-pc:/$ docker volume rm my-vol-1

my-vol-1

5.

List all the volumes

sarala@sarala-pc:/$ docker volume ls

DRIVER VOLUME NAME

**Start a container with a volume**

1.

If you start a container with a volume that does not yet exist, Docker creates the volume for you.

2.

The following example mounts the volume my-vol-1(automatically created) into home/bl/(automatically created) in the container.

docker run -d \  
 --name devtest \  
 --mount type=volume,source=my-vol-1,target=/home/bl \  
-d -h sandbox sequenceiq/spark:1.6.0 -d

docker volume inspect my-vol-1

3.

Open up a bash for the container

Container name : devtest

docker exec -i -t devtest /bin/bash

Go inside /home/bl/

Touch test.txt

Nano test.txt

Edit the content

ESC + :wq

If you run the container inside a docker machine,Check the docker machine for the files created

When you run a container inside a docker machine volumes are created inside the docker machine with in this location

Login to docker machine

<https://docs.docker.com/machine/reference/ssh/>

docker-machine ssh default

**docker@default:/$ ls**

bin home lib mnt root sys var

dev hosthome lib64 opt run tmp

etc init linuxrc proc sbin usr

**docker@default:/$ cd mnt**

**docker@default:/mnt$ ls**

sda1 sr0

**docker@default:/mnt$ cd sda1**

**docker@default:/mnt/sda1$ ls**

lost+found tmp var

**docker@default:/mnt/sda1$ cd var**

**docker@default:/mnt/sda1/var$ ls**

lib

**docker@default:/mnt/sda1/var$ cd lib**

**docker@default:/mnt/sda1/var/lib$ ls**

boot2docker docker

**docker@default:/mnt/sda1/var/lib/docker$ sudo ls**

builder containers overlay2 swarm volumes

buildkit image plugins tmp

containerd network runtimes trust

**docker@default:/mnt/sda1/var/lib/docker$ sudo ls volumes**

metadata.db my-vol-1

**docker@default:/mnt/sda1/var/lib/docker$ sudo ls volumes/my-vol-1**

\_data

**docker@default:/mnt/sda1/var/lib/docker$ sudo ls volumes/my-vol-1/\_data**

test.txt

**docker@default:/mnt/sda1/var/lib/docker$ sudo cat volumes/my-vol-1/\_data/test.txt**

hello sarala how are you?

4.

Stop the container

sarala@sarala-pc:/$ docker container stop devtest

Devtest

5.remove the container

sarala@sarala-pc:/$ docker container rm devtest

devtest

6.

Remove the volume

This is a separate step

sarala@sarala-pc:/$ docker volume rm my-vol-1

my-vol-1