**Docker Volumes**

**Recap:**

* A container is created through an image which is made up of different layers, and on top of all the layers we have a writable layer where it stores our customized data.
* But the data which was there in writable layer is temporary🡪if the container is down/stopped/restarted the data in the writable layer will be lost.
* To make present the data permanently even if the container is stopped, we use docker volumes.

**Docker Volumes:**

1. To view list of volumes available: ***docker volume ls***
2. To inspect any volume: ***docker volume inspect <volName>***

By defaut all the docker volumes will be stored at ***/var/lib/docker/volumes/<volName>/\_data*** in the host machine. If you get into the ***cd <volName>/\_data*** you can see the data which is used by the container.

1. To remove docker volumes: ***docker volume rm <volName>***
2. To remove unused/ideal docker volumes: ***docker volume prume***
3. To create a docker volume : ***docker volume create <volName>***
4. To mount/map a docker volume to a container :

***docker run -d --name <containerName> -p <hostport>:<containerPort>***

***--mount source=<hostVolPath>,destination=<containerVolPath> <image>:<version>***

OR

***docker run -d --name <containerName> -p <hostport>:<containerPort>***

***-v <hostVolPath>:<containerVolPath> <image>:<version>***

***<hostVolPath> = /var/lib/docker/volumes/<volName>***

1. Copy a docker volume from a container to another :

***docker run -d --name <containerName> -p <hostport>:<containerPort>***

***--volumes-from <container1> --name <container2>***

1. Instead of creating a volume separately with ***docker volume create <volName>*** we can make/turn a directory in host machine as a volume:

***docker run -d --name <containerName> -p <hostport>:<containerPort>***

***-v <hostVolPath>:<containerVolPath> <image>:<version>***

***<hostVolPath> = /home/ec2-user/<directory> - ec2-user’s directories are not supported as docker volumes***

***<hostVolPath> = /root/<directory> - Root user’s directories you can add as volumes***

**Docker COPY vs Volume**

* COPY and Volume both are used to transfer the data from Host machine to container.
* But with COPY, for each time when you change the file whch has been copied already in past, we need to push the latest changes to repo and need to re build the image from dockerfile and create the container to reflect the new data.
* But in Volume, once you map a directory/file as volume, whenever you made changes the volume, it updates the data at both(Host machine and Container) automatically, no need of push/re-build/new container