DOCKER CLASS-5

DOCKER VOLUMES:

- When we create a Container then Volume will be created.
- Volume is simply a directory inside our container.
- First, we have to declare the directory Volume and then share Volume.
- Even if we stop/delete the container still, we can access the volume.
- You can declare directory as a volume only while creating container.
- We can't create volume from existing container.
- You can share one volume across many number of Containers.
- Volume will not be included when you update an image.
- If Container-1 volume is shared to Container-2 the changes made by Container-2 will be also available in the Container-1.

You can map Volume in two ways:

1. Container < ------ > Container
2. Host < ------- > Container

USES OF VOLUMES:

- Decoupling Container from storage.
- Share Volume among different Containers.
- Attach Volume to Containers.
- On deleting Container Volume will not be deleted.

CREATING A VOLUME FROM DOCKER FILE:

- Create a Docker file and write

```
FROM ubuntu

VOLUME["/myvolume"]
```

- build it - docker build -t image_name .
- Run it - docker run -it - -name container1 ubuntu /bin/bash
- Now do ls and you will see myvolume-1 add some files there

- Now share volume with another Container - docker run -it - -name container2(new) - -privileged=true - -volumes-from container1 ubuntu
- Now after creating container2, my volume1 is visible
- Whatever you do in volume1 in container1 can see in another container
- touch /myvolume1/samplefile1 and exit from container2.
- docker start container1
- docker attach container1
- ls/volume1 and you will see your samplefile1

CREATING VOLUMES FROM COMMAND:

- docker run -itd --name cont1 -v /mustafa ubuntu

VOLUMES (HOST TO CONTAINER):

- Verify files in /home/ec2-use
- docker run -it - -name hostcont -v /home/ec2-user:/raham - -privileged=true ubuntu
- cd raham [raham is (container-name)]
- Do ls now you can see all files of host machine.
- Touch file1 and exit. Check in ec2-machine you can see that file.

SOME OTHER COMMANDS IN VOLUMES:

- docker volume ls
- docker volume create <volume-name>
- docker volume rm <volume-name>
- docker volume prune (it will remove all unused docker volumes).
- docker volume inspect <volume-name>
- docker container inspect <container-name>
- docker system df -v

MOUNT VOLUMES:

- To attach a volume to a container: docker run -it --name=example1 --mount source=vol1,destination=/vol1 ubuntu
- To send some files from local to container:
  - create some files
  - docker run -it --name cont_name -v "$(pwd)":/my-volume ubuntu
- To remove the volume: docker volume rm volume_name

- To remove all unused volumes: docker volume prune