(05th Dec 2025)

(Basic Docker Commands)

docker ps → this will give you list of
container ids. running

docker ps -a → this will give all
container irrespective of
their state.

docker stop <container_id> → to stop a
running
container.

docker rm <container_id> → to remove a
container.

docker images → to list images

docker rmi <image_name> → to remove image-na

↳ this also deletes all dependent container to
remove image.

docker run <image_name> → (Pull or download)
image

(Appending a command)

```
[docker run ubuntu sleep 5]
```

[Exec - execute command]

```
docker exec <container_name> cat /etc/hosts
```

You are able to execute this command inside the container.

[Attaching and detaching yourself to containers etc.]

Understand this analogy to understand this concept

Attached mode → You are sitting infront of the player, watching it display songs, seeing any errors immediately

Detached mode → the player is in another room, still playing music, but you are doing other things.

Attach → You walk into room to check on the player.

Detach → You leave the room but the player keeps running.

(Docker Demo Commands)

```
docker run -it centos bash.
```

(It ~~helps~~ ~~to~~ works like kubectle exec -it).
[You will be inside docker container]

```
docker run -d centos sleep 20
```

-d → running it in the background

sleep 20 → container comes up and sleeps for 20 sec

```
docker ps
```

→ { You will see the container there }

```
docker exec <container_id> cat /etc/release
```

⇓

we can execute inside the container

(Docker Run)

Well in docker ~~Image~~ containers we don't have interactive mechanisms.

Like for example, a program is there where if you enter your <name> then it prints "Hello <name>".

For that behaviour we have to learn about how to give input to docker.

For this to happen we have to map our standard input STDIN to docker input

```
docker run -i <image-name>
(input given here)
```

⇓ this is required.

Yet there is something missing in this, the prompt which would ask us to enter our name is not there.

{ The prompt would say, enter your name }.

That's because we are not inside the terminal of the container.

{ this is important }.

```
docker run -it <image_name>.
```

prompt will come now.

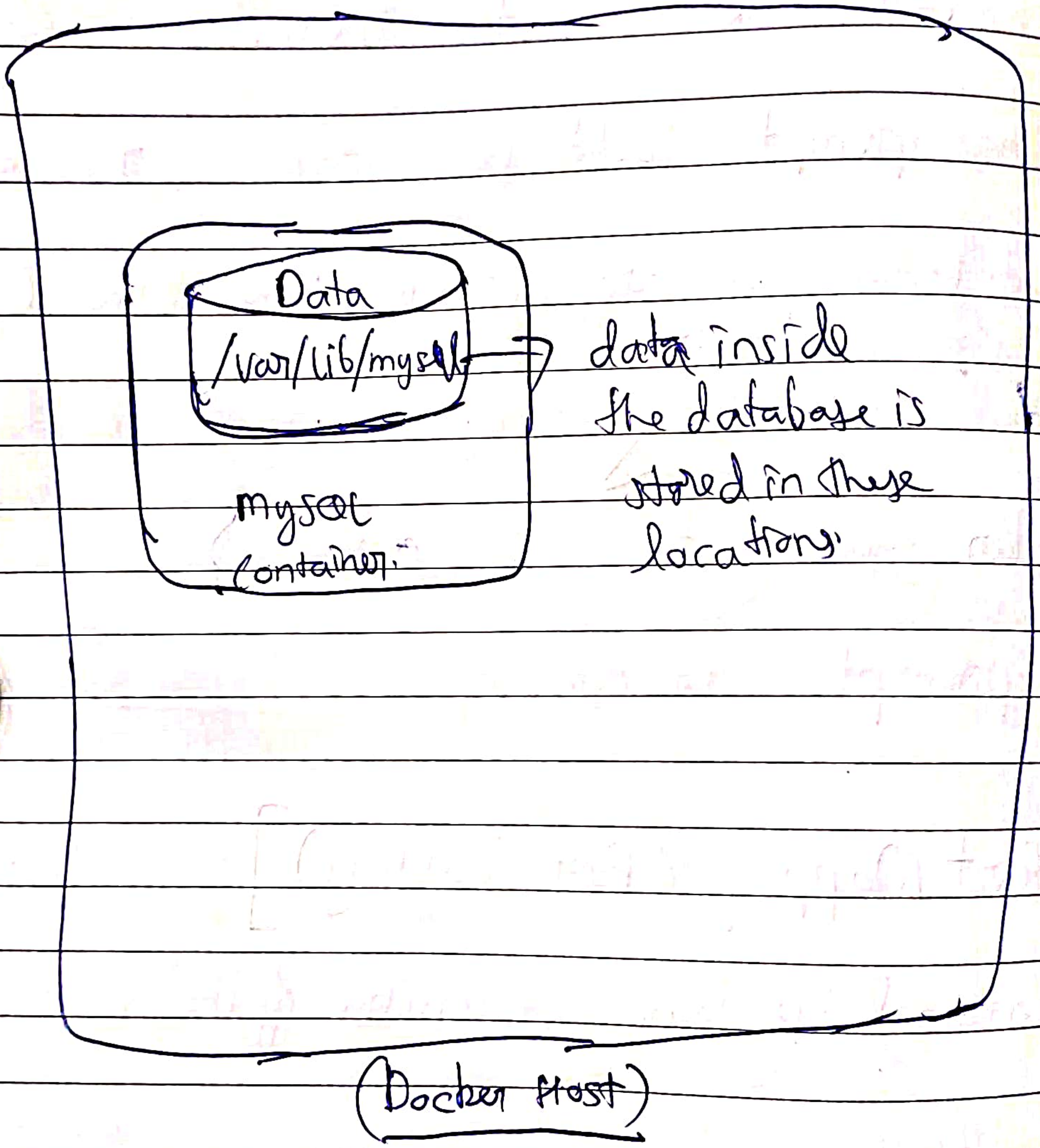
[Port Mapping (Port Publishing)]

internal ip is not accessible of the container.

```
docker run -p 80:5000 <image-name>
docker run -p 8081:5000 <image-name>
```

↓ multiple instances of the same ~~docker~~ image running

(Volume Mappings)



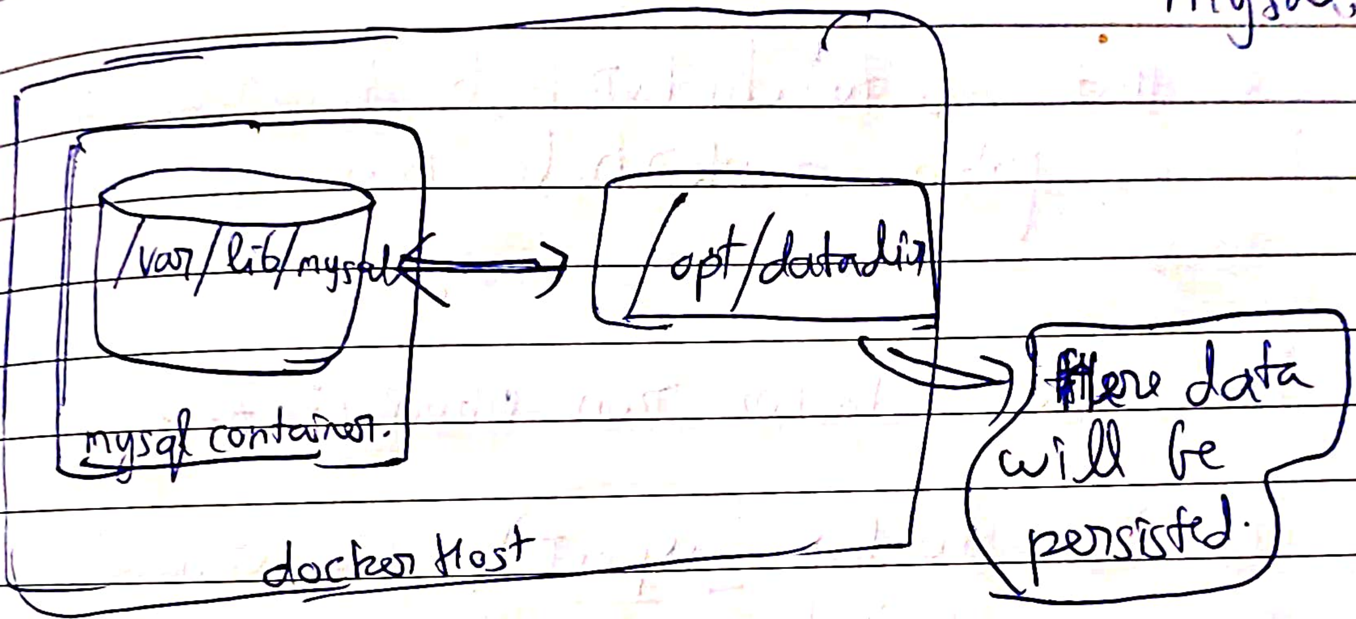
The docker container have their own isolated file system.

```
docker run mysql.
docker stop mysql.
[docker rm mysql]
```

this will remove all the data

volume mounting can be done like.

```
docker run -v /opt/datadir:/var/lib/mysql
                                    mysql.
```



<Inspect Container>

```
docker inspect <container-name>.
```

{ Shows the details }.

<Container Logs>

```
docker logs <container_name>
```

(Advanced Options Docker run)

Running Docker images of a Specific Version if we don't provide any specific version, it by default picks the latest version.

For that go to docker hub, ~~download the~~ images from docker hub have supported tags.

You can do. → docker run ubuntu:17.10

Use the docker inspect (container-name), and then scroll down to network tab to check the I.P. addresses and ports.

To port forward on your docker host.

docker run -p 8080:5000 (image)

↓

(Your docker host port)