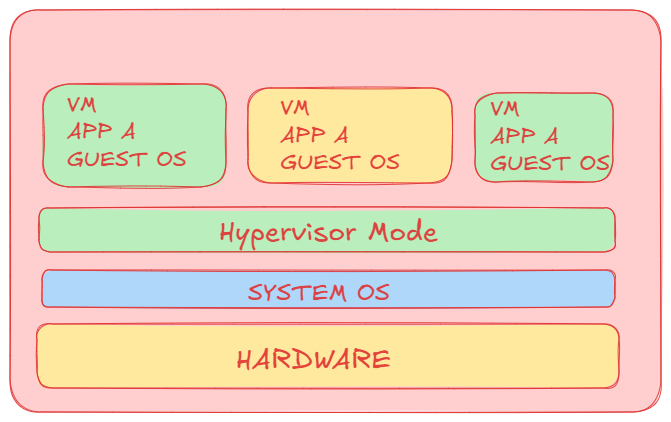
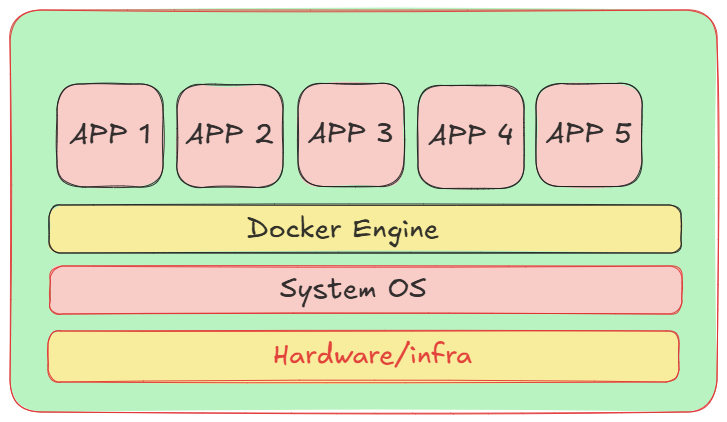
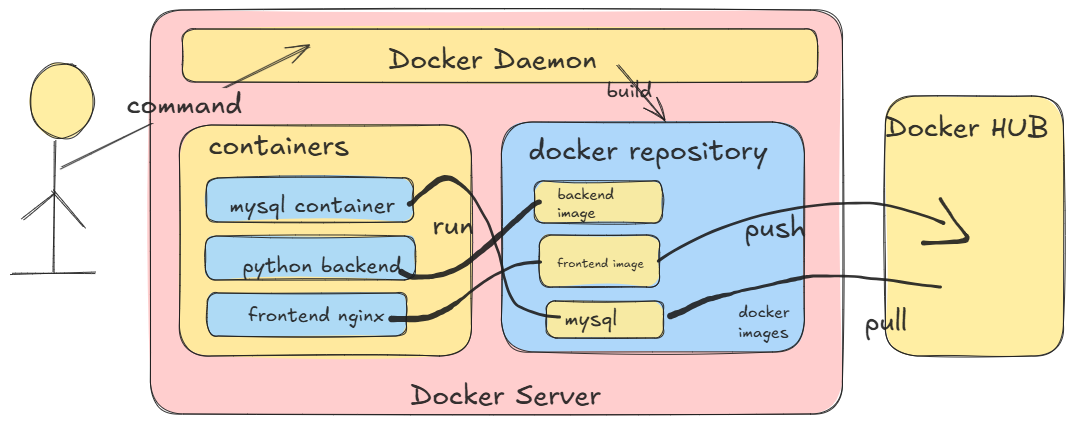
Virtual Machine



Docker



Docker is open source platform, which automates the deployment, scaling and managing application with the help of very lightweight containers.



Docker Daemon is a running thread in background, which manages the Docker instruction provided by developer.

Docker HUB: It is a central Repository, which is having all the global images available for docker.

Containers: Containers are portable applications running under docker engine.

Image: it’s a application which provides all infra which is required to set up that application. You just need to run it.

When you run docker image it creates docker container.

Example: if you want to run Jenkins Image, you no need to setup jdk as well. It will directly provided by image itself. Once its running you can see the Jenkins container running under docker containers.

**Let’s Install Docker.**

Set up Docker Apt Repo:

# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "${UBUNTU\_CODENAME:-$VERSION\_CODENAME}") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

**Install Docker packages:**

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

**Check Version**

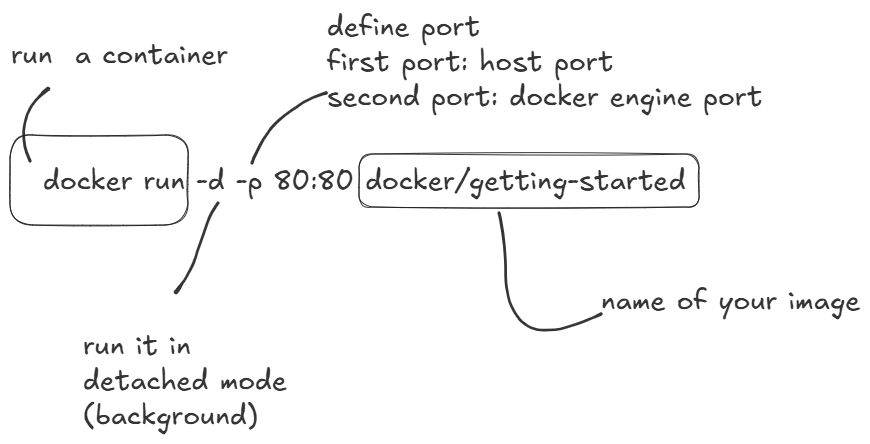
docker --version

**Then check server details:**

sudo docker version

If Docker desktop installed then just open cmd and run the commands

docker --version docker version



Here You can understand host port means your System Port

Docker engine port which is actually required by your application

Let’s say your Python-Flask application internally using port 5000 but in your system 5000 is busy.

You can expose this service at port 8080 in your system so your command will be

sudo docker run -d -p 8080:5000 python-app

In case if you want to give your name to the container use below command

sudo docker run --name myapplication -d -p 8080:5000 python-app

(my application is your containers name)

If you want to see running containers

docker ps

If you want to see all running and exited containers then

docker ps –a

Exited containers you can directly remove:

sudo docker rm container-name

if container is running then you can’t remove directly you have to stop them first and then you can remove

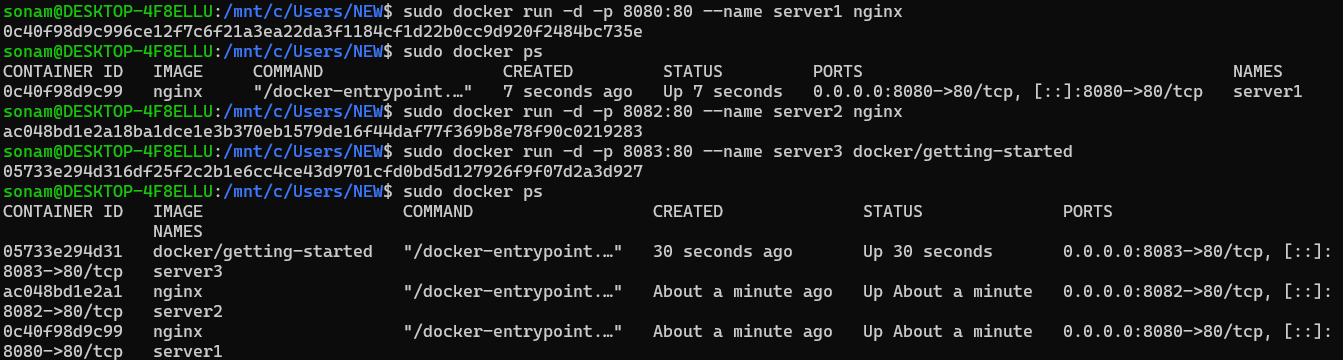
sudo docker stop container-name

sudo docker rm container-name

Yes, you can directly try to remove forcefully using –f but it’s not a suggested way.

sudo docker rm container-name -f

If the container name is given though its in exited state then also you can’t reuse the same container name. If you want to use them make sure you remove that container in exited state.



At a time 3 service exposed internally which is using port 80 but exposed differently in your system on port 8080,8082, 8083

So you can check output

<http://localhost:8083/>

<http://localhost:8082>

<http://localhost:8080>

Let’s Run MySQL Container

sudo docker run --name mysql-server -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql

to access the container:

sudo docker exec -it mysql-server mysql -u root –p

(here –it means interactive mode)

Mysql-server is the my of my container

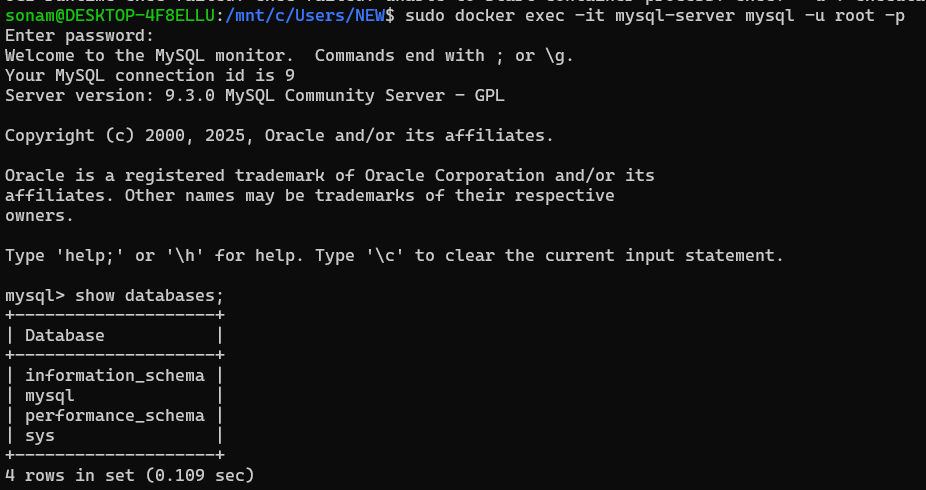
Mysql is the command I want to execute

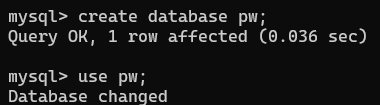
-u (username is root)

-p

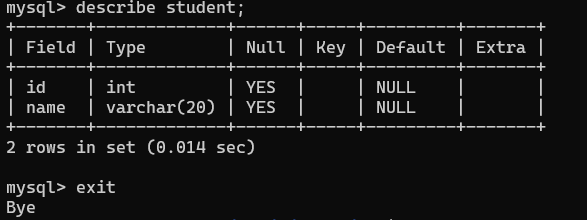
It will promp to enter password: enter password, which you have passed as an environment variable when you have created container and you are inside the mysql terminal.

Where you can run all mysql commands.





create table student (id int, name varchar(20));



Creating Own Images in Docker

For creating images, you have to write your project related Docker file.

In this file you have to provide the instruction what to do with your project, how to set up files, what is the starting point, which port to expose.

Your file name is Dockerfile without any extention.

**Very Basic Terms for writing Dockerfile:**

FROM:

It is to specify BASE Image like use Ubuntu or use any other os.

FROM Ubuntu:20.04

RUN:

Executes commands in a new layer on the top of your current image, installing packages.

RUN apt-get update && apt-get install python3 –y

CMD:

Provides default for executing containers

CMD[“python3”,”app.py”]

EXPOSE:

Inform your docker to start the service to the mentioned port

EXPOSE 8080

ADD:

Copy files, directories or remote locations files

ADD xyz /etc/config

COPY

copy only local files

COPY . ./app

ENTRYPOINT

Configures your container excutable

ENTRYPOINT [“npm”,”run”,”dev”]

WORKDIR

Set working directory

WORKDIR /app