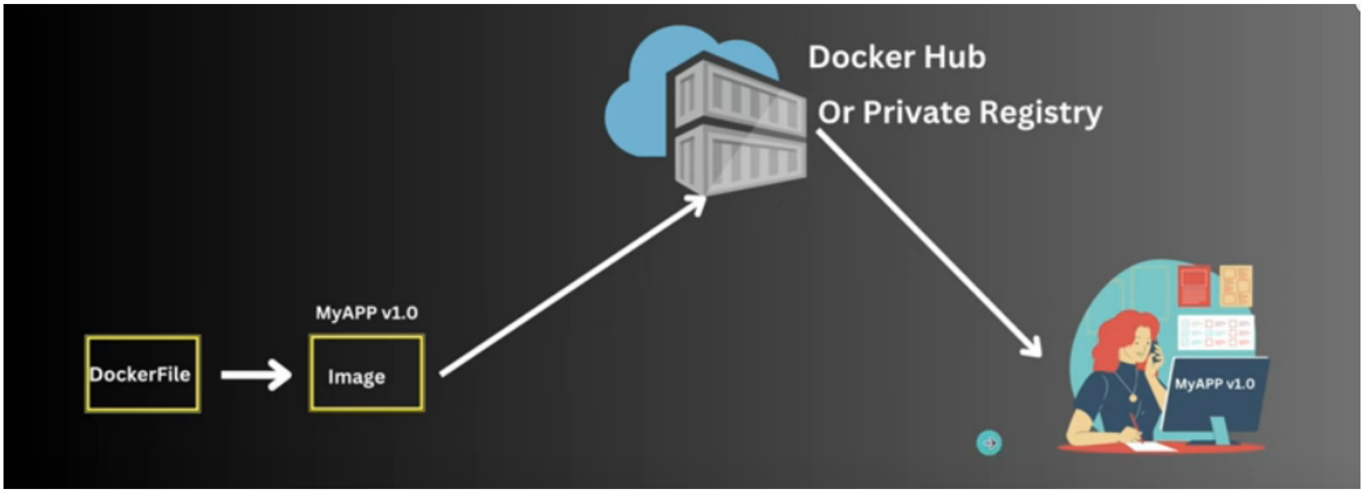
It is open source platform enables to package applications all with dependencies into containers.

Main Components :
Docker File : It is simple text file with instructions to build an image.
Docker Image : Single file all dependencies and libraries to run the program.
Docker Container : run image on different system.it will act as or run a container on different system.
Docker Registry : It is Central repository for storing and distributing image.



Docker commands :

```
docker --version

docker run nginx
docker images
docker run -d images //running in detached mode /background
docker ps
docker ps -a
docker stop <container_name>
docker start <container_name> //once created start anytime
docker remove <container_name> //need to stop first
docker run -d --rm --name nc nginx //remove without stoping and container name is nc.Once we have stopped it will remove automatically.
```

```
PS E:\Docker_TUTO> docker run -d --rm --name nc -p 8080:80 nginx
c90a6863799d56f742cf2fa08e734f8696afcf0978563a0dd827179dfb4356ba
PS E:\Docker_TUTO> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c90a6863799d nginx "/docker-entrypoint.…" 5 seconds ago Up 5 seconds 0.0.0.0:8080->80/tcp nc
```

bind container ip with localhost to access it outside the container.

```
docker logs nc
docker logs -f nc //live logs
```

```
docker attach nc //running in foreground
```

container small env where run this application

```
docker exec -t nc bash
PS E:\Docker_TUTO> docker exec -t nc bash
root@c90a6863799d:/#
//inside the container
```

Create custom Container

create DockerFile in your Project

DockerFile

```
FROM node
WORKDIR /app

COPY ..
RUN npm install
ENV PORT 3000
CMD ["node","app.js"]
```

.dockerignore

```
node_modules
```

Now build own image

```
docker build -t email:01 . //where dockerfile is stored
docker images
```

```
docker run -d --rm --name myapp -p 3000:3000 email:01
docker ps
```

```
docker run -d --rm --env PORT=3030 --name myapp -p 8080:3030 email:01
docker logs myapp
```

see on localhost:8080

the main problem is when we stop the docker container then all data will be lost to prevent this use docker volume.

Docker Volume

```
docker run -d -v myapp:/app/ --rm --env PORT=3030 --name myapp -p 8080:3030 email:01
```

preserve data

```
docker volume rm myapp
```

Bind Mount Docker Volume : data should store on localhost then it should be mount to docker volume

```
docker run -d -v :/app/email.txt --rm --env PORT=3030 --name myapp -p 8080:3030 email:01
```

Docker Registry: Go to docker hub create new repo
it gives new images name so need to create this new image with this name for that use :

```
docker build -t philippaul/email:01 .
```

check image created or not

for push to docker hub

```
docker login
docker pushphilippaul/email:01
```

user can get this application here by

```
docker pull philippaul/email:01
```

Docker Compose : user has to write lengthy command at every time to run image
for that
docker-compose.yml in which version image,container name ports,volume ,env given and stored in project directory

after that only run using

```
docker-compose up -d
docker ps
docker compose down
```

Docker Network : project has multiple container like frontend ,bacend,database container so need to store it in network

for that :

```
docker network create my-net
docker network ls
```