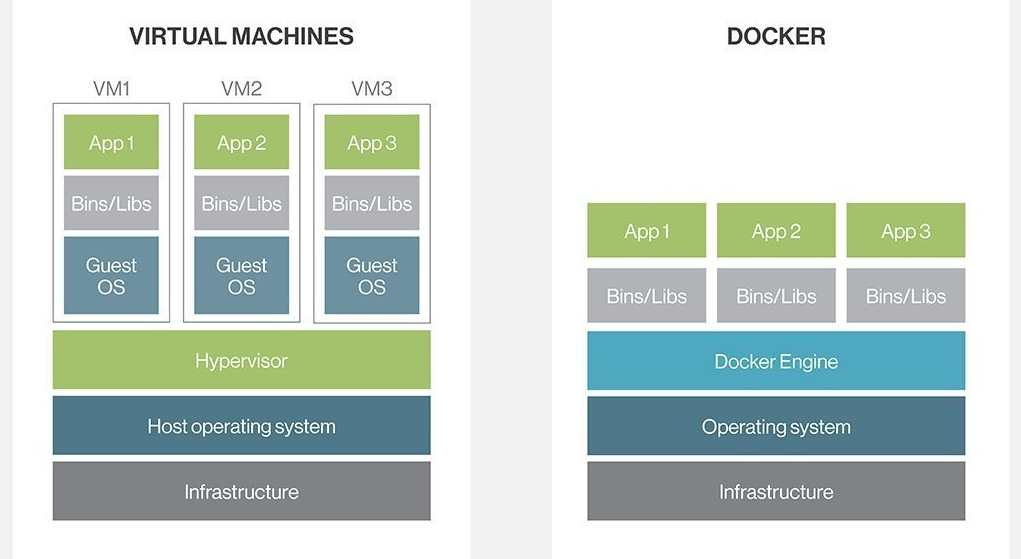
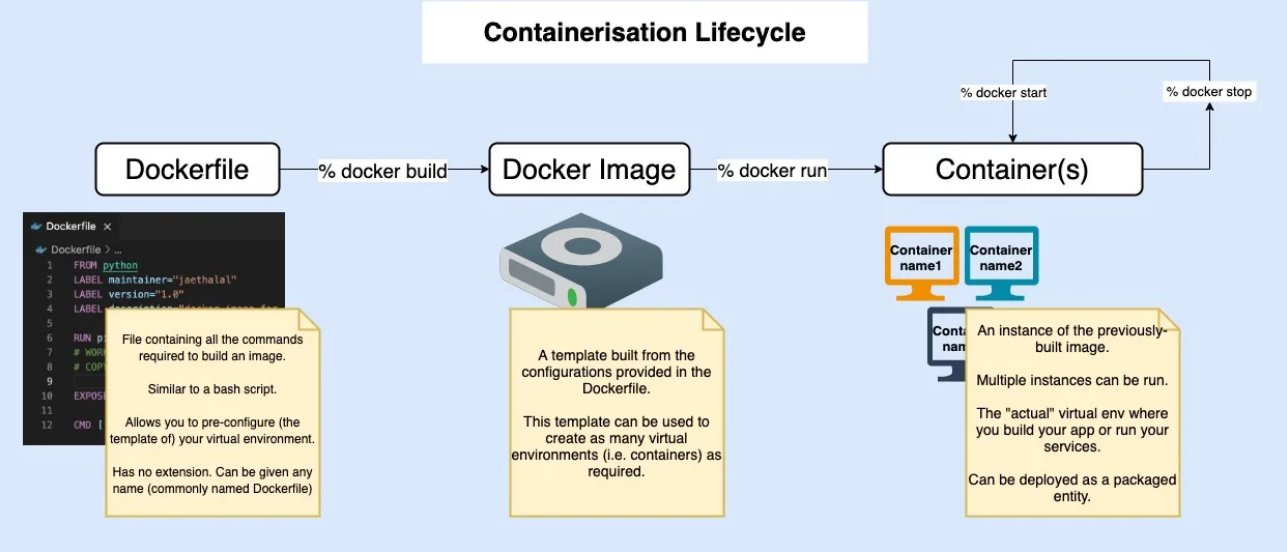
**Docker -** A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing.



**Docker image** - contains application code, libraries, tools, dependencies and other files needed to make an application run.

**Docker containers** - are the live, running instances of Docker images

**Docker container lifecycle**



**Dockerfile** – it contains commands to create docker images

**Dockerfile Commands**

FROM – to pull the base image

RUN – to execute commands

CMD – Specifies the default command to run when a container is started from the image.

ENTRYPOINT – Specifies the executable to run when the container starts

WORKDIR – to set the working directory

COPY – to copy a files/directories from docker machine to docker container

ADD – to copy files/folders and also from docker machine to docker container and but also allows you to fetch files from remote URLs and extract compressed files.

EXPOSE – to open container port

ENV – to set environmental variables

jar -cvf projectname.war \* - to create a war file

**Docker commands**

docker build -t <imagename>. – to build docker image

docker run -itd -p 9000:8080 --name <container name> <image name> - to run container from

docker images - to check container images

docker ps - to check running container docker ps -a –to check all the available containers(stopped running)

docker stop $(docker ps -a -q) - stop all the running containers

docker rm $(docker ps -a -q) – delete all the containers

docker rmi $(docker images -a -q) – delete all the image

**Docker network(optional)**

docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 098db155d1bb – to check ip addres of container

docker network ls - to check network in docker engine

docker network create my-network – to create

docker network rm my\_network - to remove