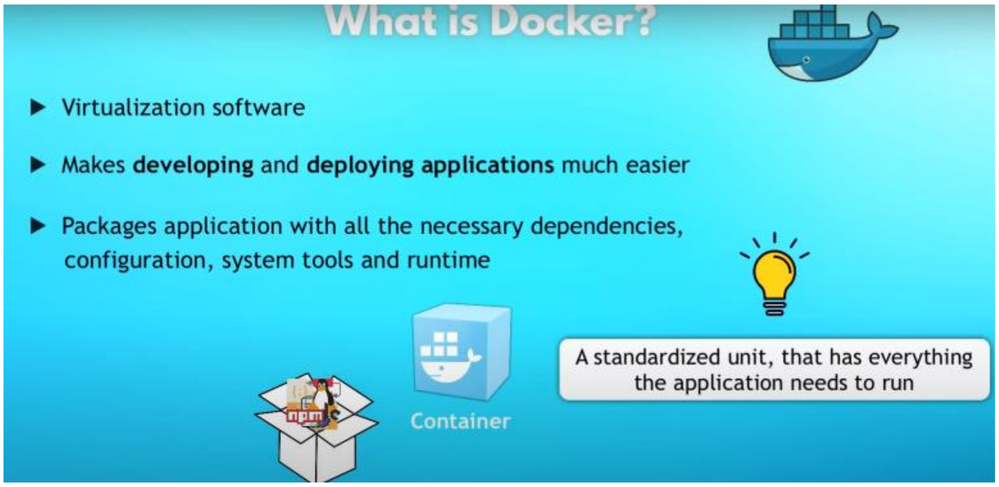
DOCKER

Monday, October 9, 2023 11:23 PM

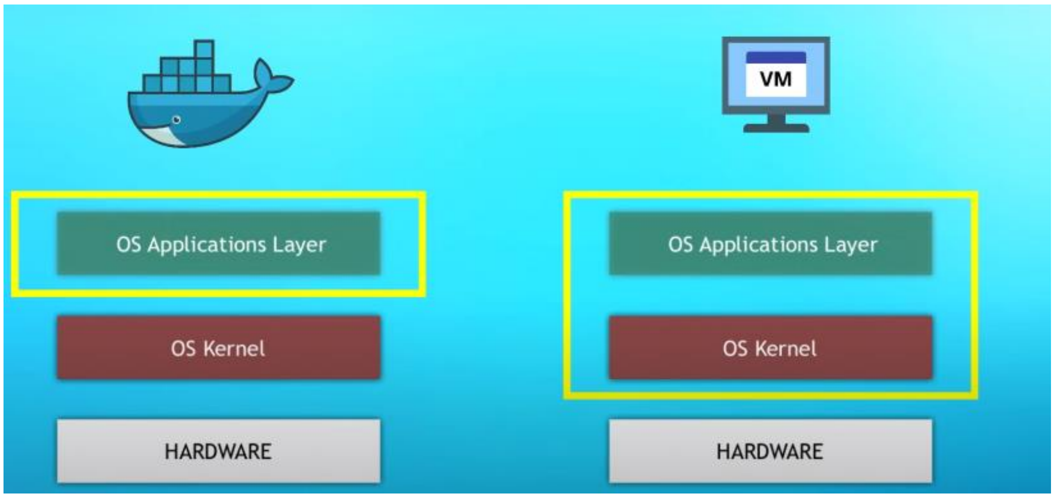


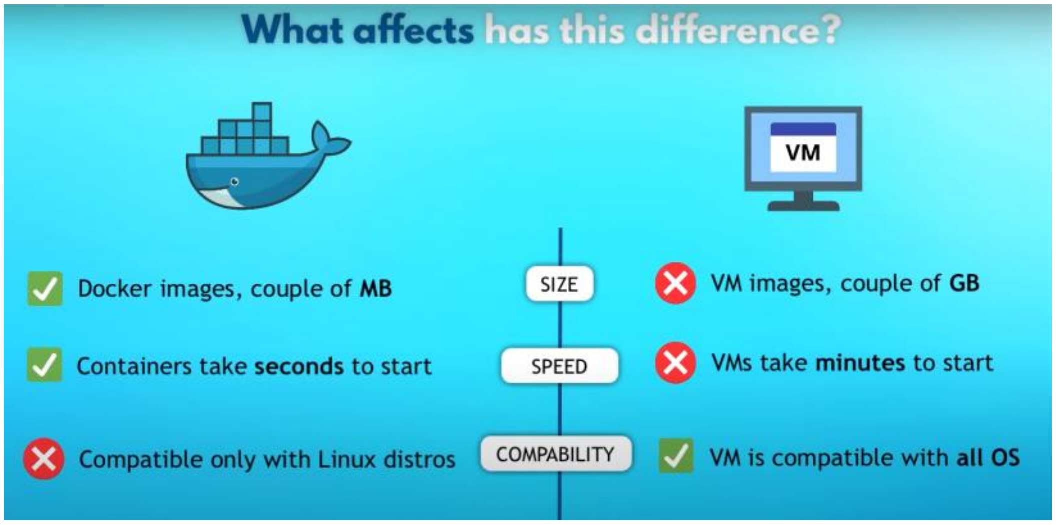
What is Docker?

- Virtualization software
- Makes **developing** and **deploying applications** much easier
- Packages application with all the necessary dependencies, configuration, system tools and runtime

Container
A standardized unit, that has everything the application needs to run

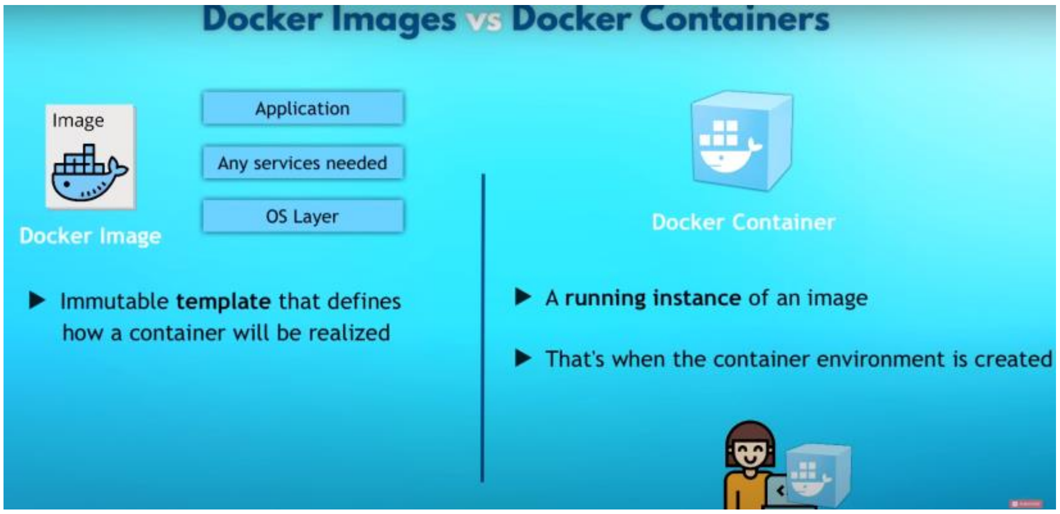
DOCKER VS VM





What affects has this difference?

Docker		VM
✓ Docker images, couple of **MB**	SIZE	✗ VM images, couple of **GB**
✓ Containers take **seconds** to start	SPEED	✗ VMs take **minutes** to start
✗ Compatible only with Linux distros	COMPABILITY	✓ VM is compatible with **all OS**



Docker Images vs Docker Containers

- Immutable **template** that defines how a container will be realized

- A **running instance** of an image
- That's when the container environment is created

Commands:

To fetch images:
->Docker images
To fetch containers:
->Docker ps
To pull image:
->>docker pull name:version
(default version is latest)
Ex: docker pull nginx:1.13
To run image:
->>docker run name:version
Ex: docker run nginx:1.13
To run image detaching from terminal(doesn't close if terminal closed):
->docker run -d name:version
Ex: docker run -d nginx:1.13
To get logs of container:
->docker logs containerID
To run image without pulling locally:
->docker run imageName:version
To stop a container:
->docker stop containerID

To do PORT BINDING(-p or -publish):
->docker run -d -p hostPort:containerPort imageName:version
Ex: docker run -d -p 9000:80 nginx:1.13

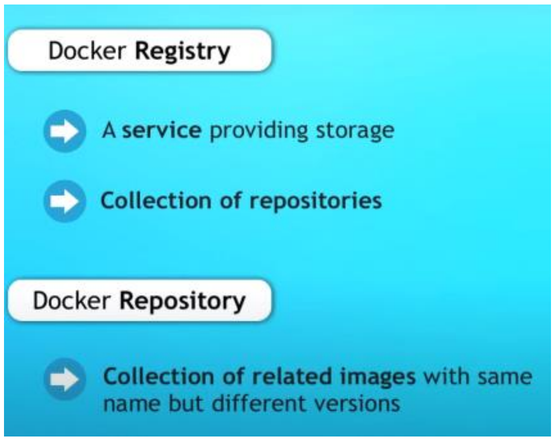
Note: only one image can run on a port at given time instance

To get list of all containers created whether running or not:
->docker ps -a

To start a container previously created and was stopped:
->docker start containerID

To run a container by providing a name:
docker run --name {name} -d -p {hostPort}:{containerPort} {imageName}:{version}

DOCKER REGISTRY VS REPOSITORY



Docker Registry

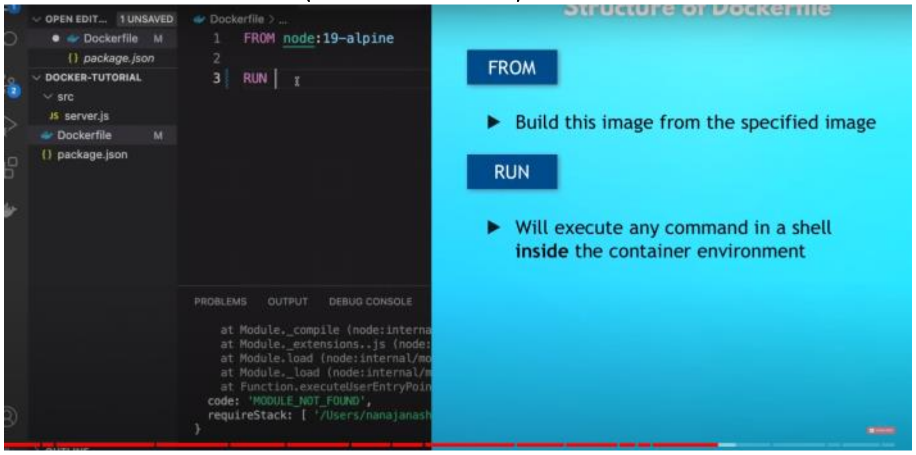
- A service providing storage
- Collection of repositories

Docker Repository

- Collection of related images with same name but different versions

BUILDING YOUR OWN IMAGE AND HOST ON DOCKER

->WRITE YOUR CODE
->CREATE A DOCKER FILE(USE BASE IMAGE)



```
FROM node:19-alpine

RUN
```

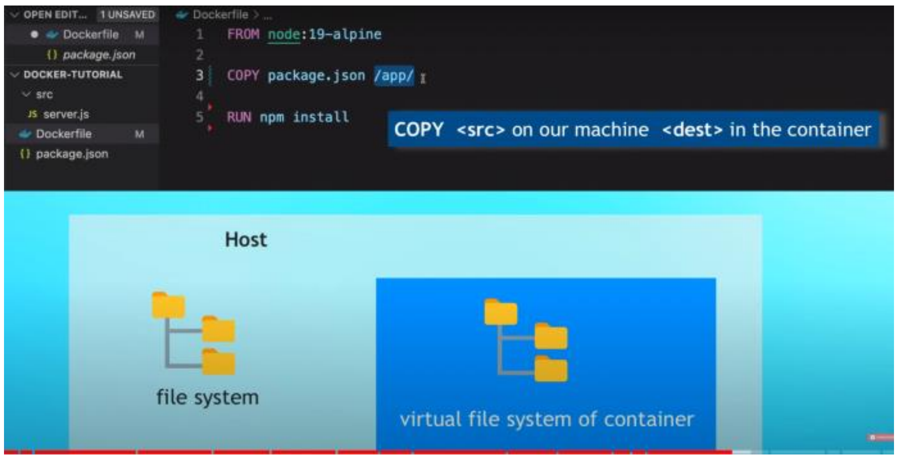
```
at Module._compile (node:interna
at Module._extensions..js (node:
at Module.load (node:internal/mo
at Module._load (node:internal/m
at Function.executeUserEntryPoin
code: 'MODULE_NOT_FOUND',
requireStack: [ '/Users/nanajanash
}
```

FROM

- Build this image from the specified image

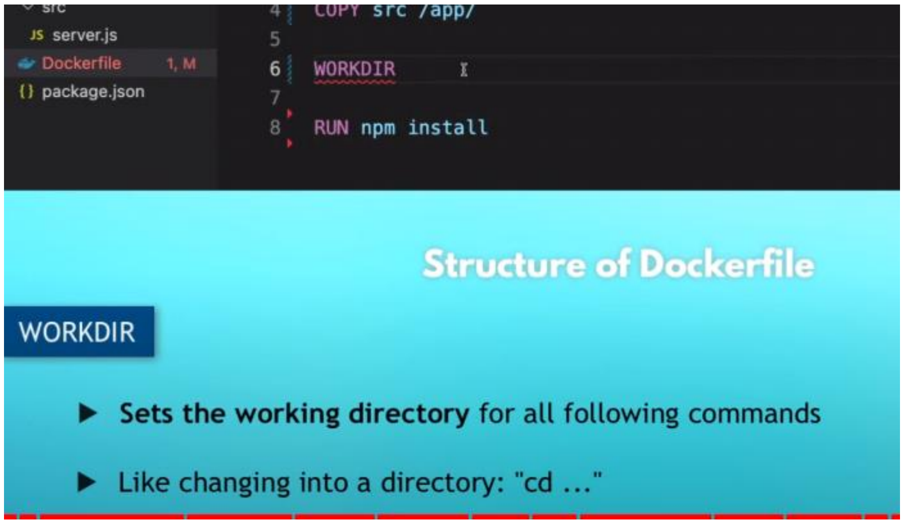
RUN

- Will execute any command in a shell inside the container environment



```
FROM node:19-alpine

COPY package.json /app/

RUN npm install
```

COPY <src> on our machine <dest> in the container

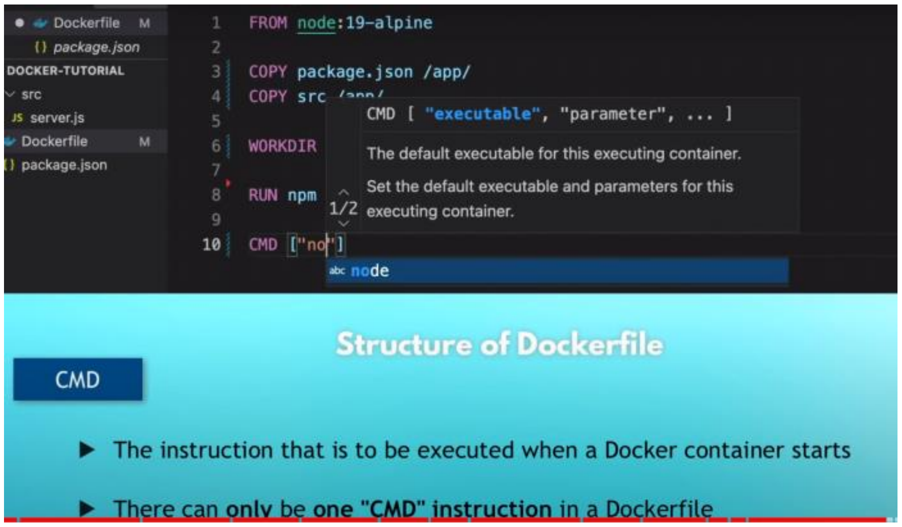


```
COPY src /app/

WORKDIR

RUN npm install
```

Structure of Dockerfile

WORKDIR

- **Sets the working directory** for all following commands
- Like changing into a directory: "cd …"



```
FROM node:19-alpine

COPY package.json /app/
COPY src /app/

WORKDIR

RUN npm

CMD ["no"]
```

```
CMD [ "executable", "parameter", ... ]
```

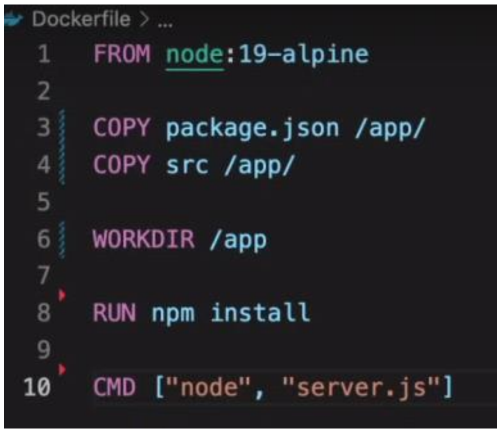
The default executable for this executing container.

Set the default executable and parameters for this executing container.

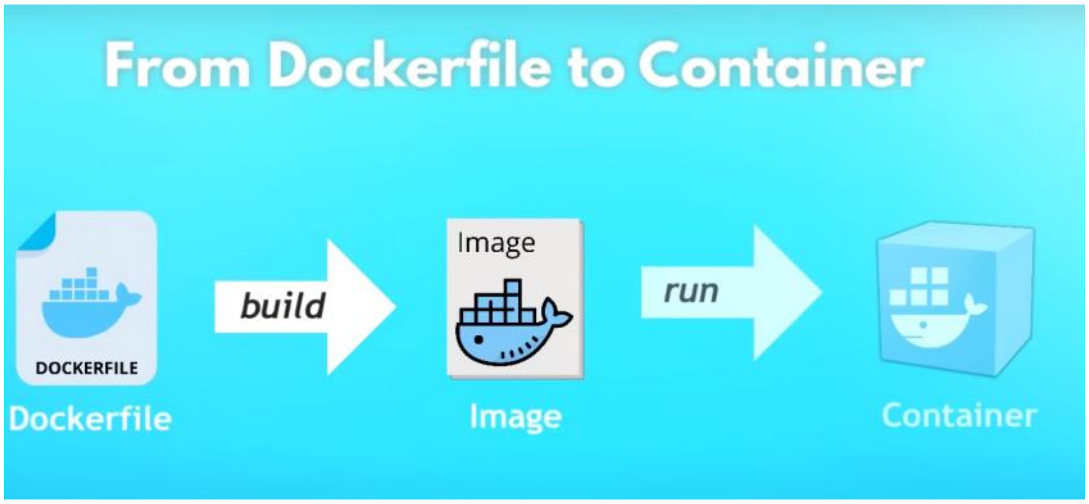
Structure of Dockerfile

CMD

- The instruction that is to be executed when a Docker container starts
- There can **only** be **one "CMD" instruction** in a Dockerfile



```
FROM node:19-alpine

COPY package.json /app/
COPY src /app/

WORKDIR /app

RUN npm install

CMD ["node", "server.js"]
```



Now its time to build docker image from dockerFile:

->Docker build -t {imageName}:{imageVersion} {location of dockerFile from which image to be build}