**DOCKER**

**What is Docker?**

It is **virtualization software** that is used to **build, deploy and run** the applications easily.

Docker packages the applications with all necessary dependencies, libraries, and system tools together into the **container**.

A **container** is a standardized unit that has everything required to run an application. - an isolated environment

Can run different versions on the same OS

It is portable, easily shared and distributed.

**BEFORE**

Each developer needs to install and configure directly on their OS.

The installation process would be different for each OS

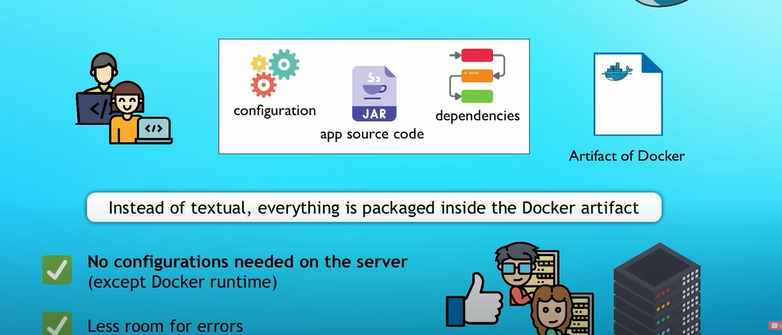
Many steps - are prone to errors.

**VM vs Container**

Virtualises everything, virtualises only the OS application layer

Faster, lightweight.





Docker comes with - docker engine, docker CLI and GUI

Docker Engine is an open source containerization technology for building and containerizing your applications. Docker Engine acts as a client-server application with:

* A server with a long-running daemon process [dockerd](https://docs.docker.com/engine/reference/commandline/dockerd).
* APIs which specify interfaces that programs can use to talk to and instruct the Docker daemon.
* A command line interface (CLI) client [docker](https://docs.docker.com/engine/reference/commandline/cli/).

**Docker Images:**

It is an executable application artifact

It is a blueprint to the docker container

It includes applications source code, but also the complete environment configuration,

Appl - JS

Env - linux

Service - node , npm

Add env variables, create directories, files etc for your appn.

**Docker Container:**

Actually starts the image

It is the running instance of the image

One image can run on multiple containers.

**Docker registry**

A storage and distribution system for docker images

It finds and shares dicker images

DOCKER HUB is the biggest docker registry available.

**TAGS**

Every app has upgrading and updation. They are versioned and are found under tags.

Tag : latest - refers to te newest version.

**PORT Binding**

Application runs inside the docker’s isolated network.

Need to expose to the host

Finds the containers port to the host port to make the service available for the outside world,

**Docker file**

**A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. /var/lib/docker**

**Images are always read only. any change made requires build again.**

**COMMANDS**

Docker images

Docker ps

Docker ps -a

Docker pull name:tag [docker pull nginx:1.23]

Docker pull nginx (brings the latest tag)

Docker run name:tag [docker run nginx:1.23] starts a new container

Docker run -d name:tag (ignores the background running and returns only the ID)

Docker logs container id(shows the working now)

Docker run -d -p 3000:80 nginx:1.23

Docker start id(does not create a new container and the old changes stay!)

Docker stop id

Docker exec -it id/name /bin/bash (to make changes within the backend files)

Docker run –name [petname] name:tag(to change the name of the container)

To build image: docker build -t [name:tag] [location]

To run and to build a container: docker run -d -p 4000:4000 - -name [dummyname] [imagename:tag]