Presentation on Docker

Meer murtaza

Student (Department of Computer Science)

FAST-NUCES PESHAWAR

CONTENT:

- Software Deployment
  - Large scale software deployment
  - Virtualization
  - Containerization
  - Intro to Dockers
  - Dockers Installation
  - Docker Commands
  - Containerizing an App

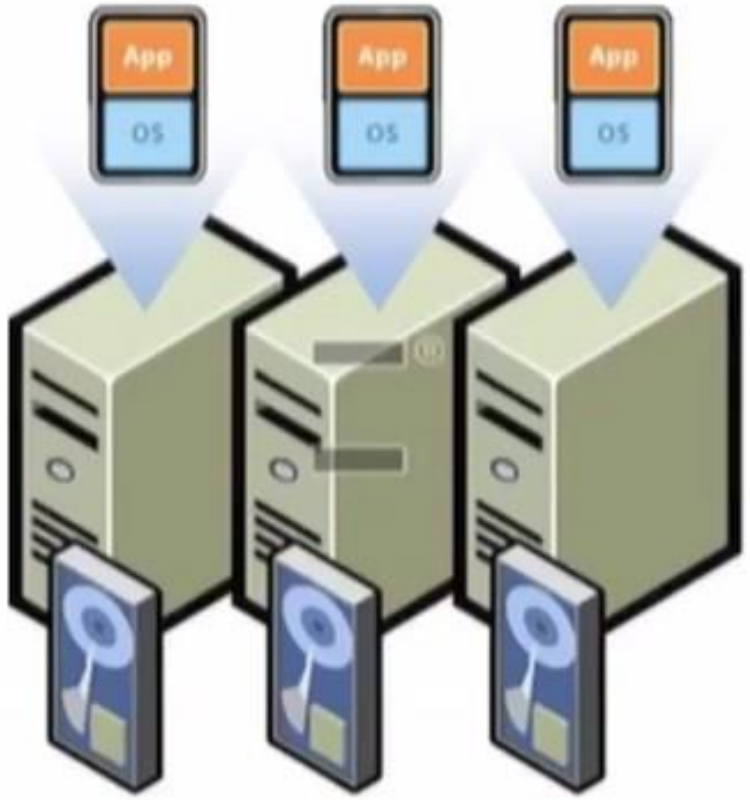
SOFTWARE DEPLOYMENT

IN PAST

Usually One App and one server rule.

- **Reasons were**
  - Unable to judge resources
  - Different Infrastructure and dependencies

- **Disadvantages**
  - Very Costly
  - Resource Wastage
  - Many Servers to manage



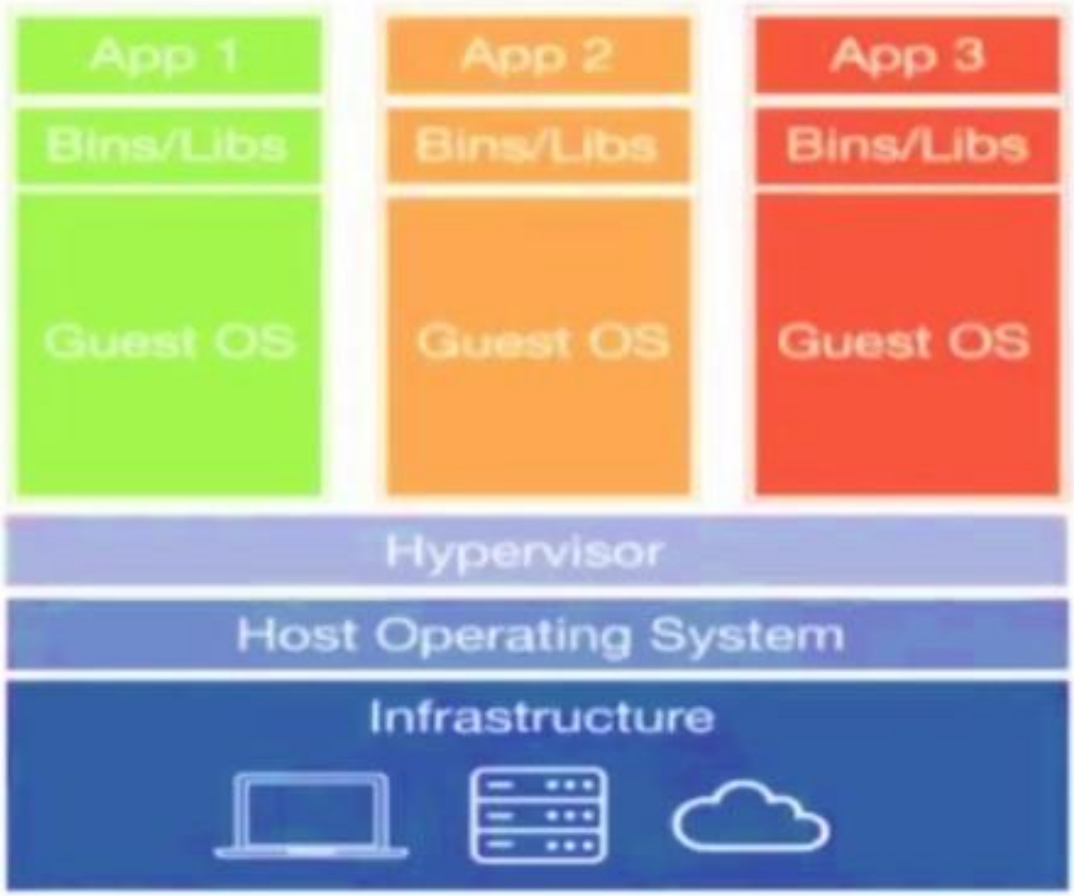
VIRTUALIZATION

VIRTUALIZATION

VMware in 1998….

- **Come with Virtualization**
  - Multiple App on a Single Server
  - Different OS and dependencies on the same server using VMs
  - Better than old one
  - Saves resources

- **Disadvantages**
  - OS consume a lot of resources.
  - Licensing Cost of OS

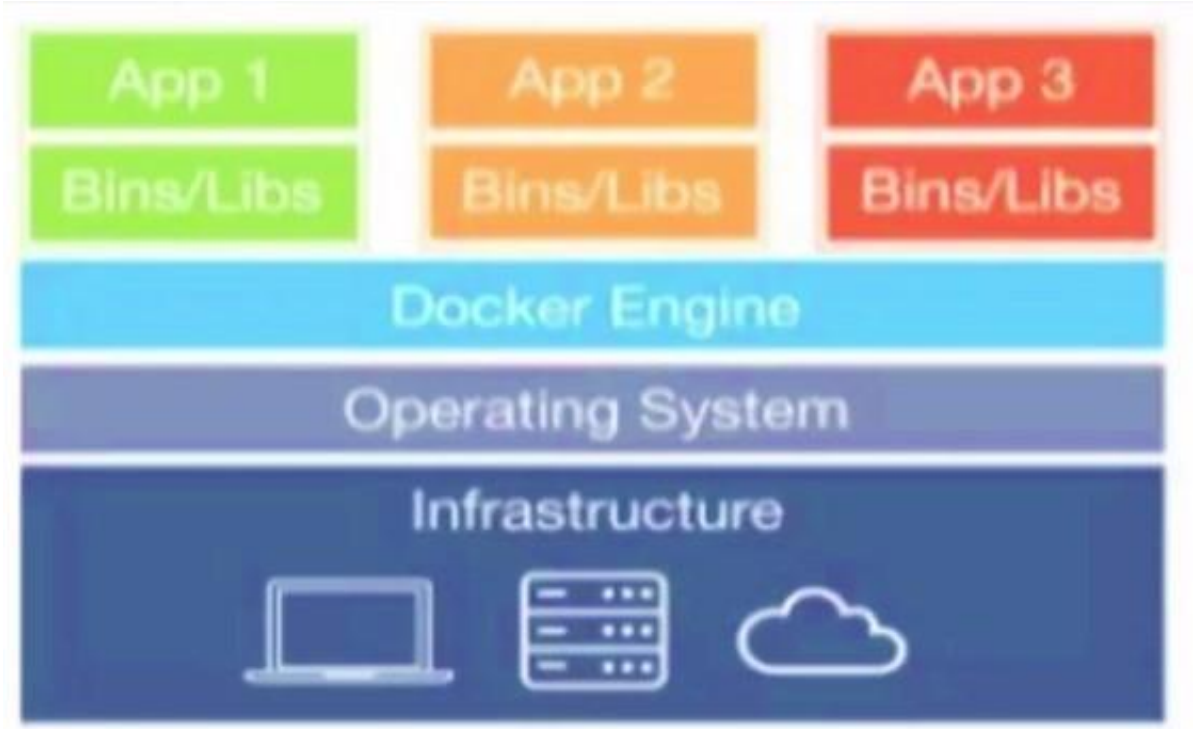


CONTAINERIZATION

CONTAINERIZATION

Containerization is a method to package an application so it can be run, with its dependencies, isolated from other processes.

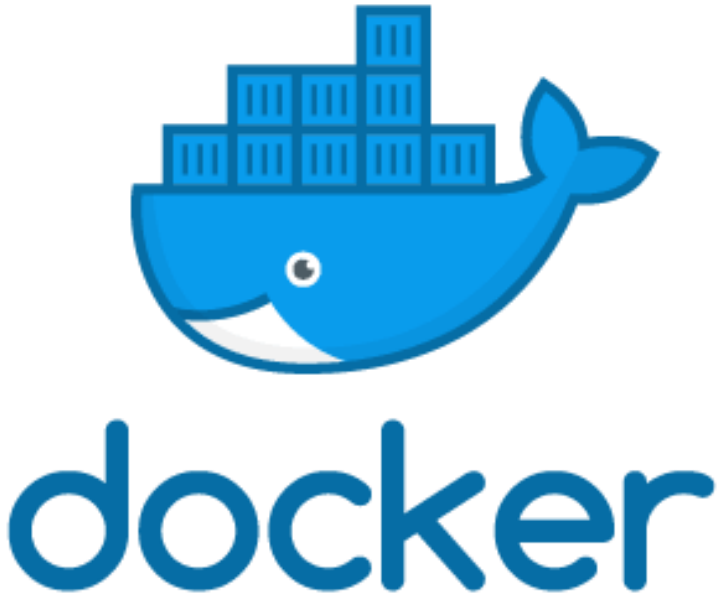
- **Diff between Container and Virtual Machine**
  - Single OS
  - Less Hardware resource



DOCKER

DOCKER

- Docker is not a container.
- Docker is a software that manages containers.
- Docker engine is a core software that runs and manages containers.
- A platform for building, running and shipping applications.

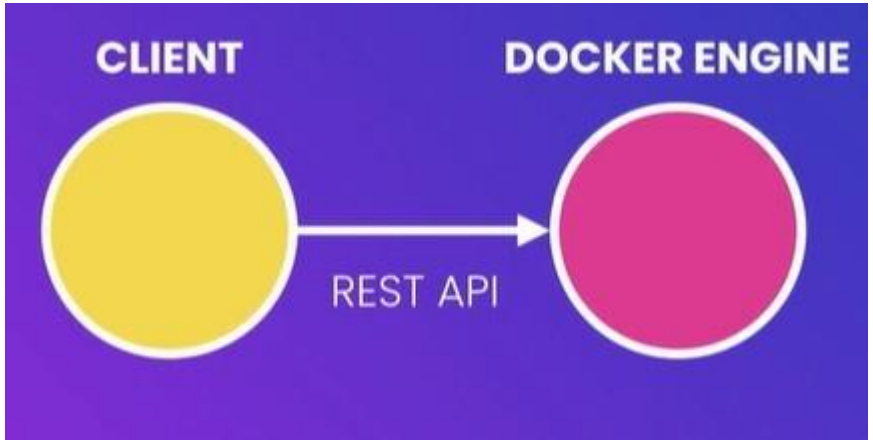


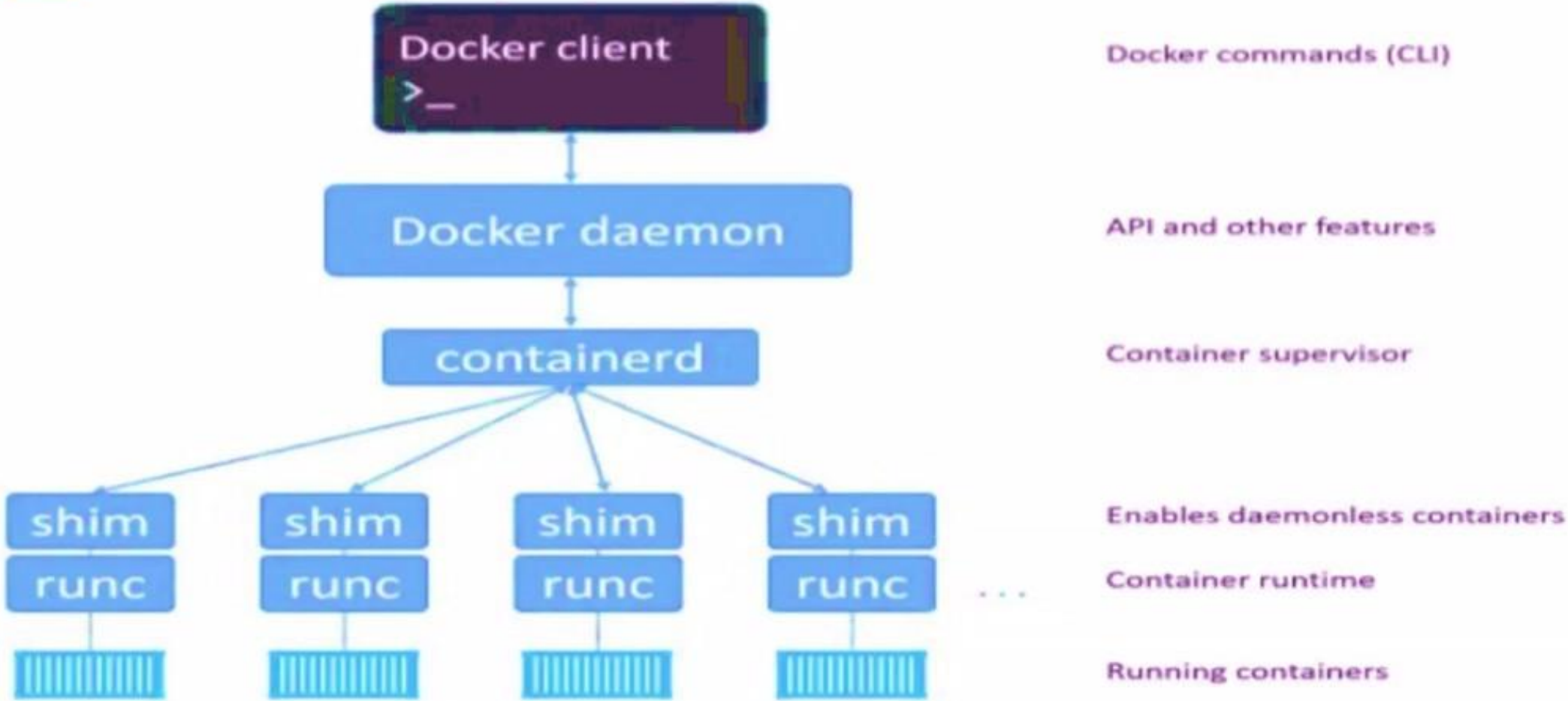
DOCKER ARCHITECTURE

DOCKER ARCHITECTURE

DOCKER ARCHITECTURE

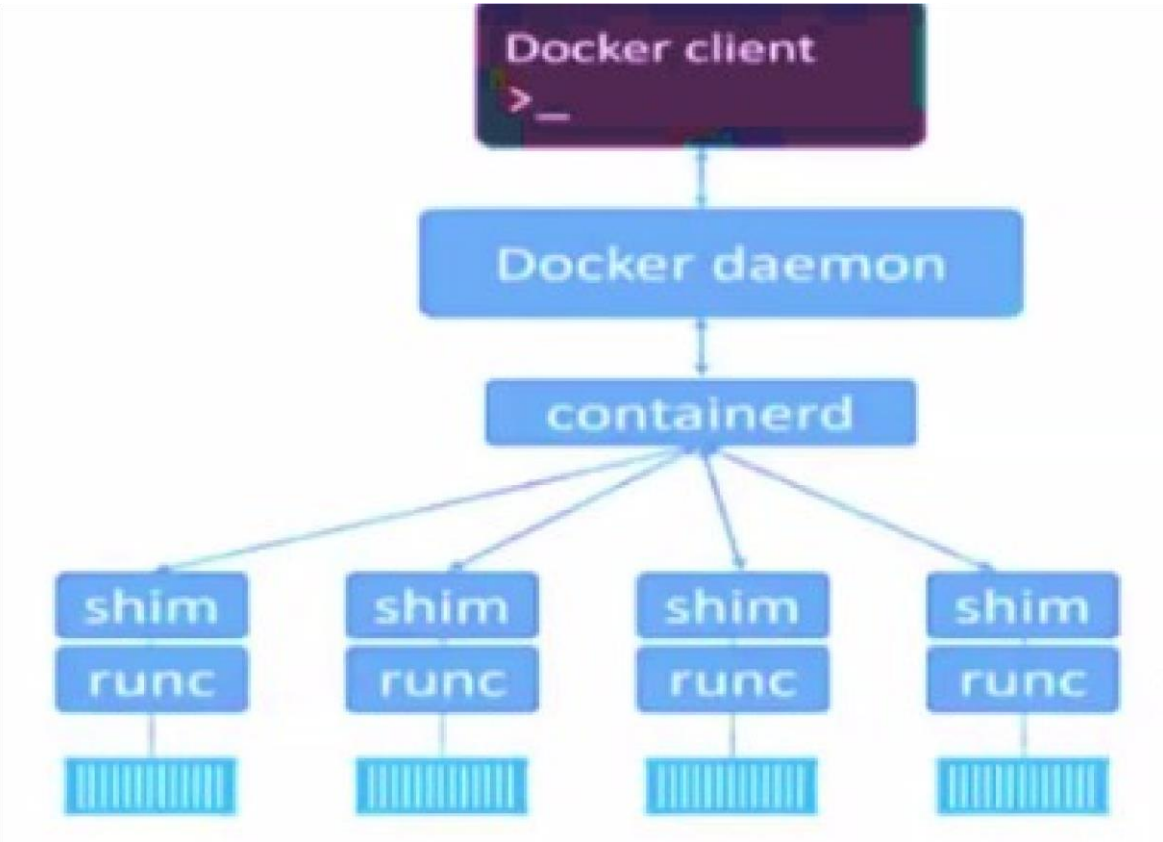
- Docker uses client/server architecture. It has a client component that talks to the server using a RESTful API.
- The server is also called the Docker engine (or daemon) runs in the background and is responsible for doing the actual work.




Docker client
>_
Docker daemon
containerd
shim
runc
shim
runc
shim
runc
shim
runc
. . .
Docker commands (CLI)
API and other features
Container supervisor
Enables daemonless containers
Container runtime
Running containers

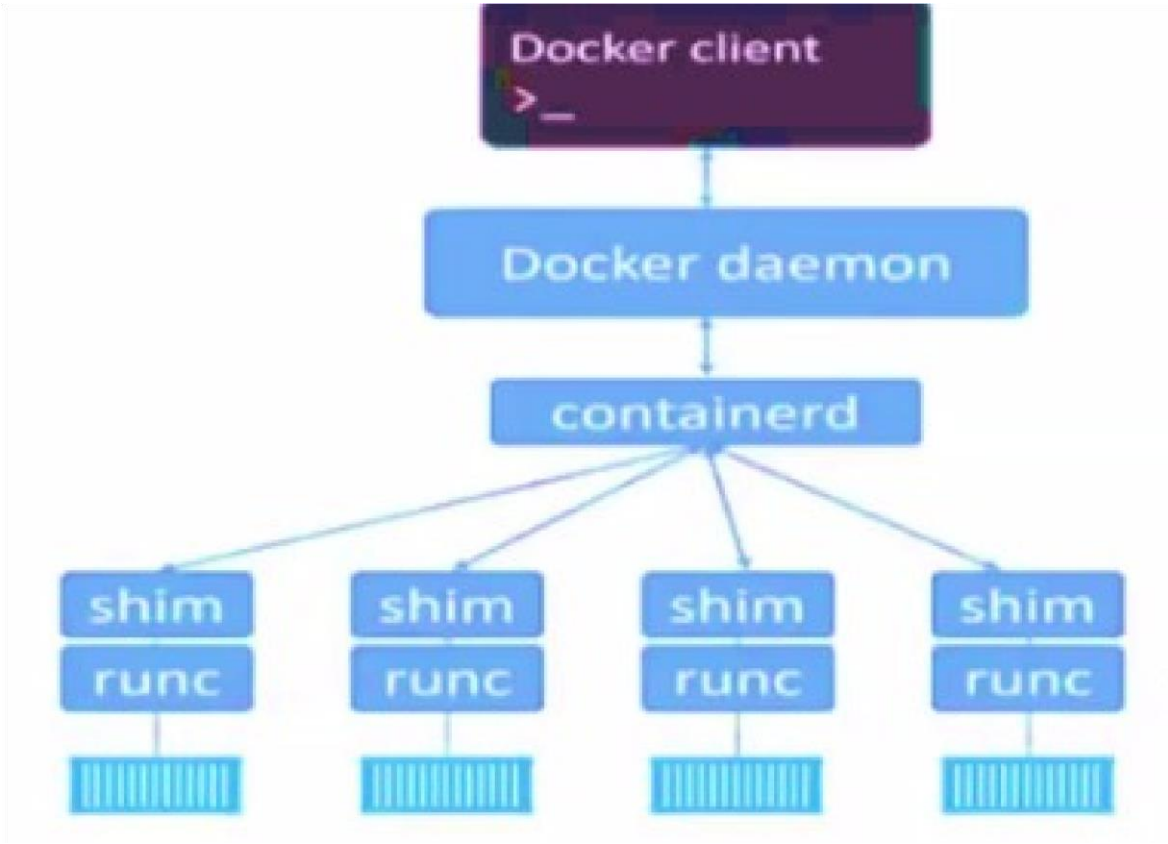
DOCKER ARCHITECTURE

- **Docker Daemon**
  - Docker Daemon listens for API requests and manages docker objects such as images, network, volumes.



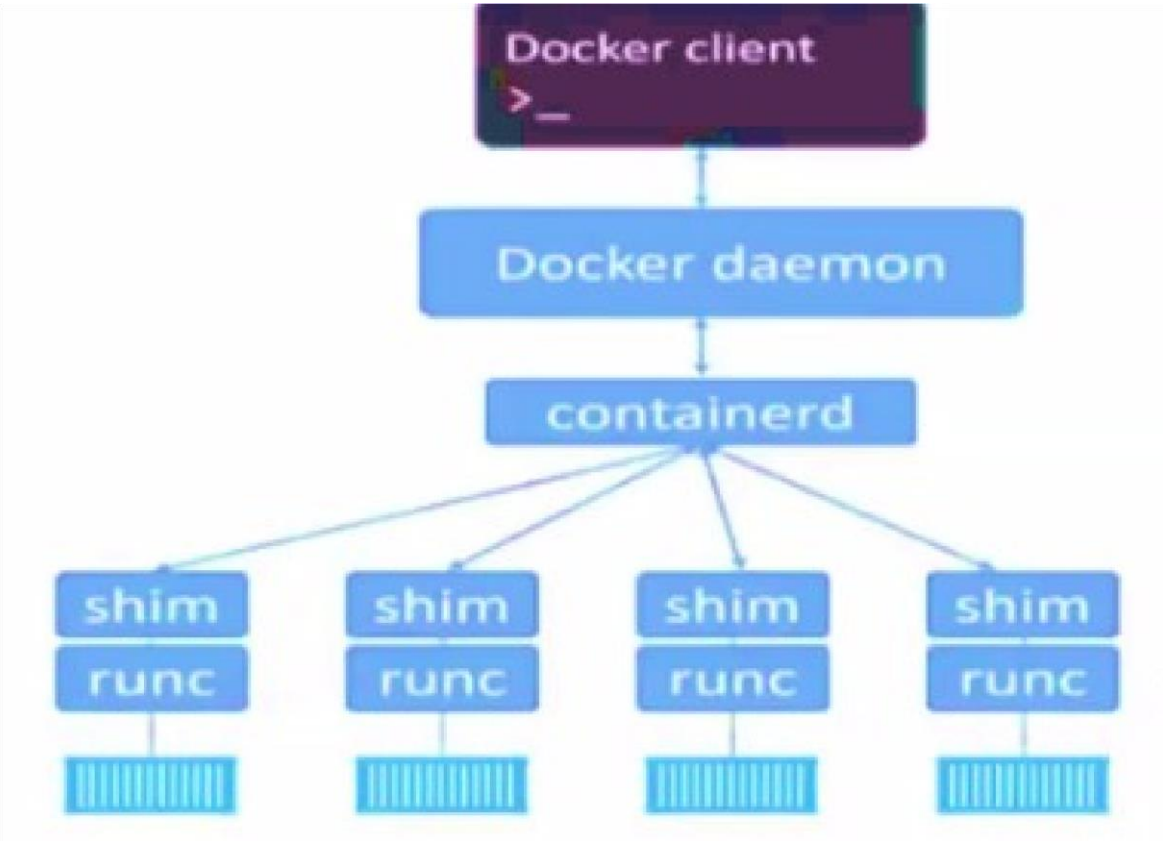
DOCKER ARCHITECTURE

- **Containerd**
  - Bridge between daemon and runc.
- It helps in
  - Starting and Stopping Containers
  - Pausing and unpausing
  - Destroying containers



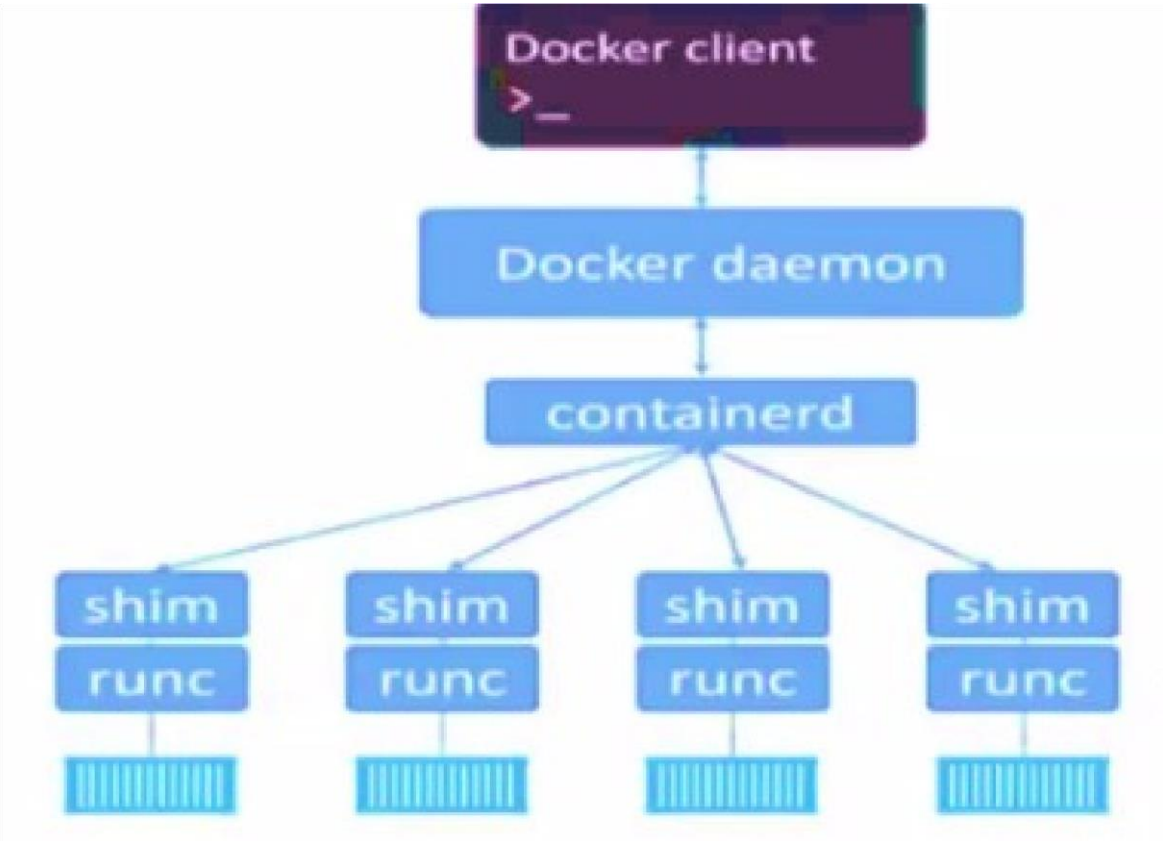
DOCKER ARCHITECTURE

- **Runc**
  - Runc often refer as a container runtime.
  - To create containers.



DOCKER ARCHITECTURE

- **Shim**
  - Used for the implementation of daemonless containers.
  - Shim makes it possible to maintain and upgrades without impacting running containers.
    - (no need to stop and kill containers)



DOCKER INSTALLATION

INSTALLATION STEPS

- **Install Docker**
  - sudo apt-get install docker-ce docker-ce-cli containerd.io

DOCKER COMMANDS

- docker -- version
- docker info
- docker version

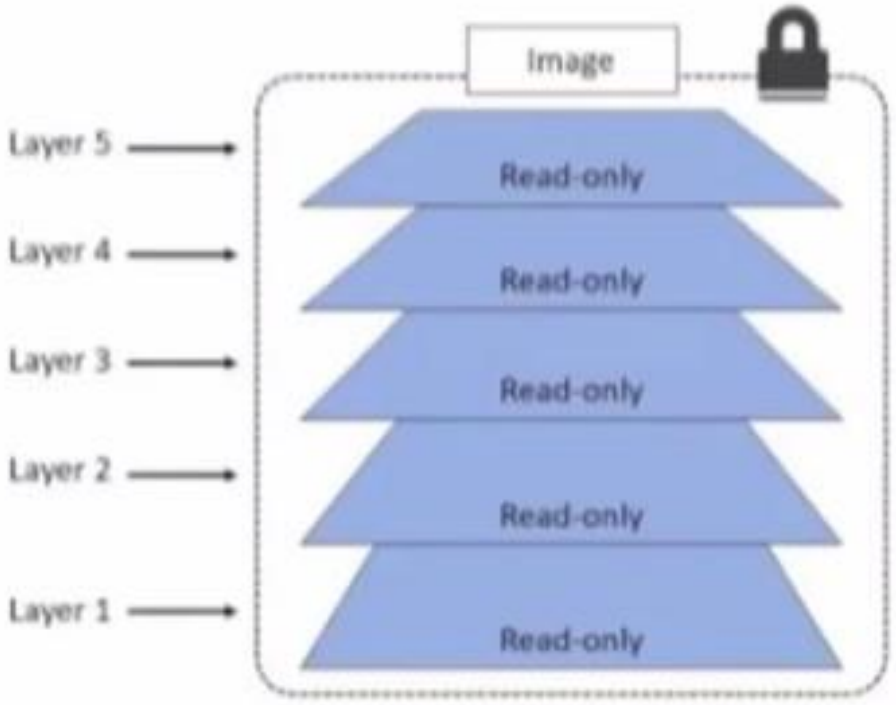
IMAGES

DOCKER IMAGE

DOCKER IMAGE

Image

- A container image is a lightweight, standalone, executable package of software that includes everything needed to run an application.
  - Code
  - Runtime
  - System Tools
  - System Libraries
  - Settings
- Images become containers when they run on Docker Engine.
- Images are made up of multiple layers.



Image

- Inside of the image is A cut-down operating system (OS), and all of the files and dependencies required to run an application.
- In this way, each layer contains different things required to run a containerized app.
- Common layers among different images are downloadable only once and get reuse in all images.
- We build containers based on images and that is why images are sometimes called stopped containers.
- We can create images from actual stopped containers.

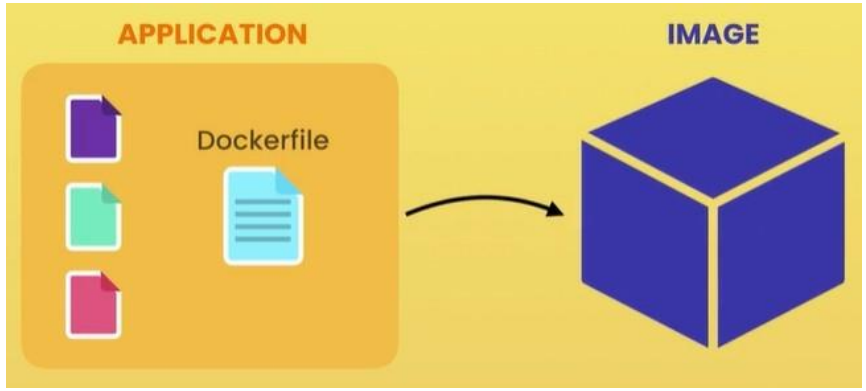
DOCKER IMAGE

BUILDING IMAGE

Build Image

Image

- Once the container is up and running made from an image, the two constructs become dependent on each other and you cannot delete the image until the last container using it has been stopped and destroyed.
- The purpose of container is to run an application.
- However, containers are all about being fast and lightweight.
- Official Ubuntu Docker image which is about 120 MB.



- docker build -t first-docker-app .

Image Listing

- docker image ls / docker images

- **Removing Image**
  - docker image rm first-docker-app:latest

We cannot delete an image until the last container using it has been stopped and destroyed.

Docker File has two main purposes

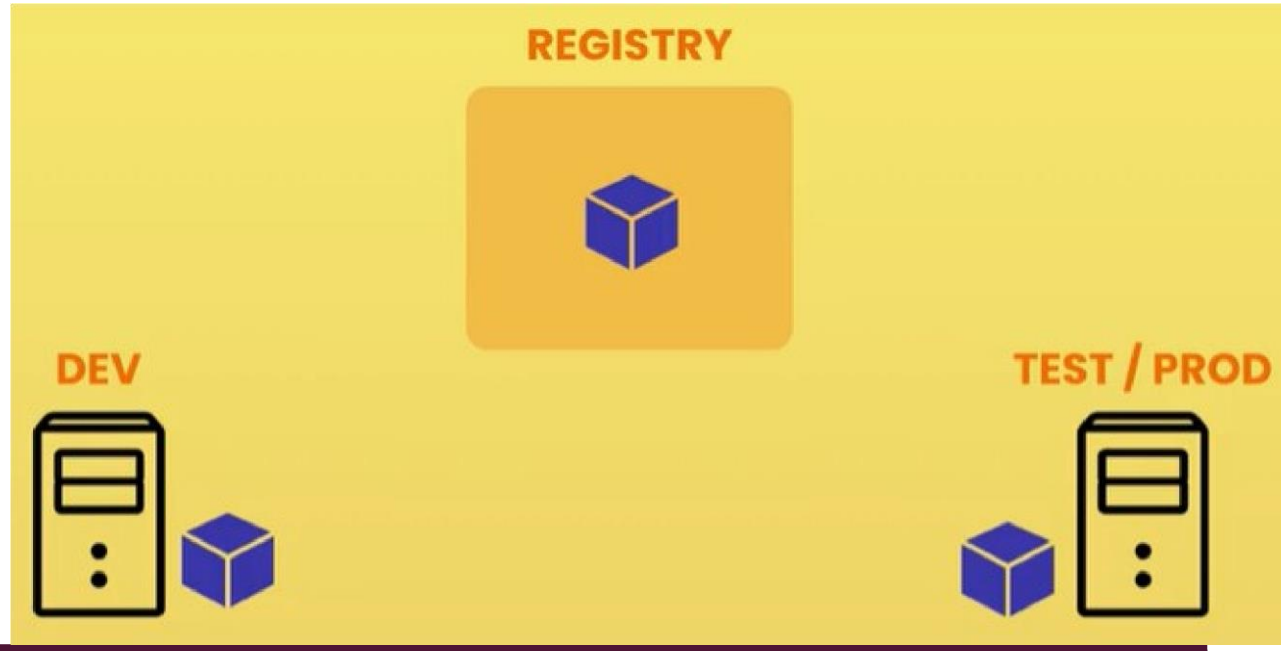
- To describe an application
- To tell Docker how to containerize the application

Note: Images are used for creating containers.

IMAGE REGISTRIES

- Docker Images are portable.

- Docker images are stored in image registries, the most common registry is Docker Hub. (https://hub.docker.com)
- Image registries contains multiple repositories.
- Image repositories contains multiple images.
- Docker hub has a concept of official and unofficial registries.



PUSH IMAGES TO DOCKER HUB

```
docker push meermurtaza/first-docker-app:latest
```

Error

The push refers to repository [docker.io/meermurtaza/first-docker-app]

An image does not exist locally with the tag: meermurtaza/first-docker-app

docker tag first-docker-app meermurtaza/first-docker-app

docker push usama9876/first-docker-app:latest

PULL IMAGES FROM DOCKER HUB

- Addressing images from official repositories
  - Docker image pull <repository> : <tag>

- E.g. docker image pull ubuntu : 18.0

- Addressing images from unofficial repositories
  - Docker image pull username/<repository> : <tag>
  - E.g. docker image pull meermurtaza/abc : latest

- If you do not specify an image tag after the repository name, docker will assume that you are referring to the image tagged as latest.

RUNNING A CONTAINER

RUNNING A CONTAINER

- A container is the runtime instance of an image.
  - docker run -it meermurtaza/helloworld sh
  - docker run -d meermurtaza/helloworld
    - docker run -d –p 5000:80 meermurtaza/helloworld

RUNNING A CONTAINER

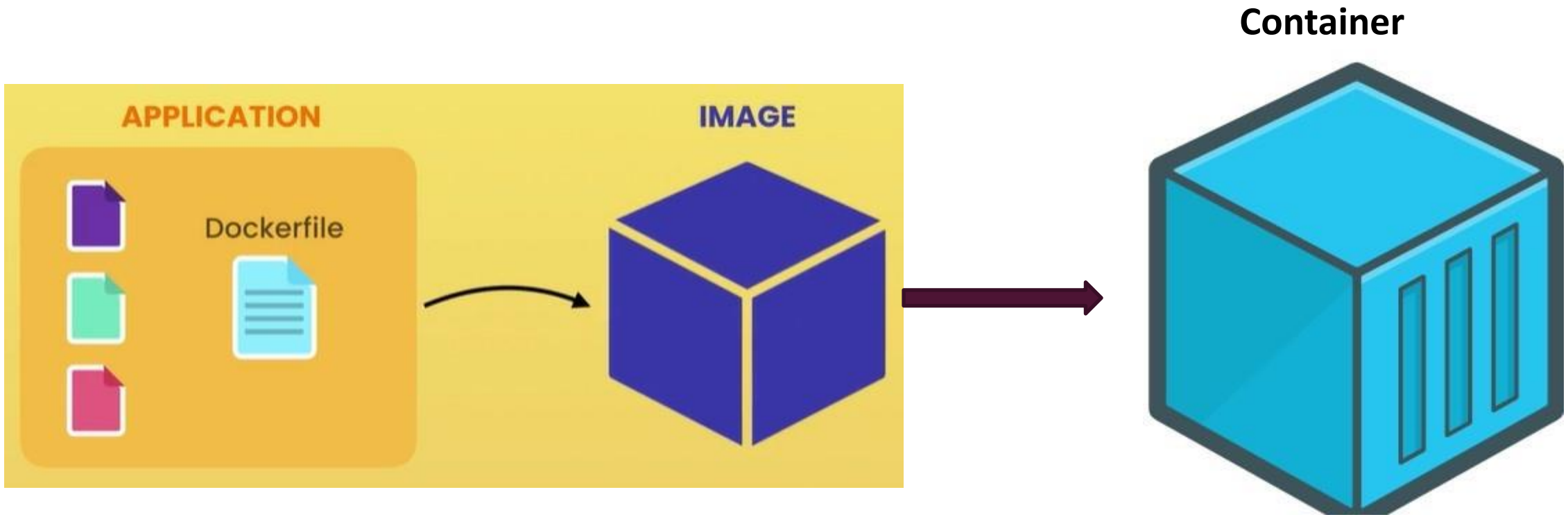
Some more commands.

- docker container ls / docker ps
- docker container ls –a / docker ps –a
- docker container start container name/id
- docker container stop container name/id

- docker container rm container name/id
- exit vs ctrl+pq
- docker exec -it container name

CONTAINERIZING A SIMPLE HTML-JS APP FROM SCRATCH

CONTAINERIZING AN APP FROM SCRATCH

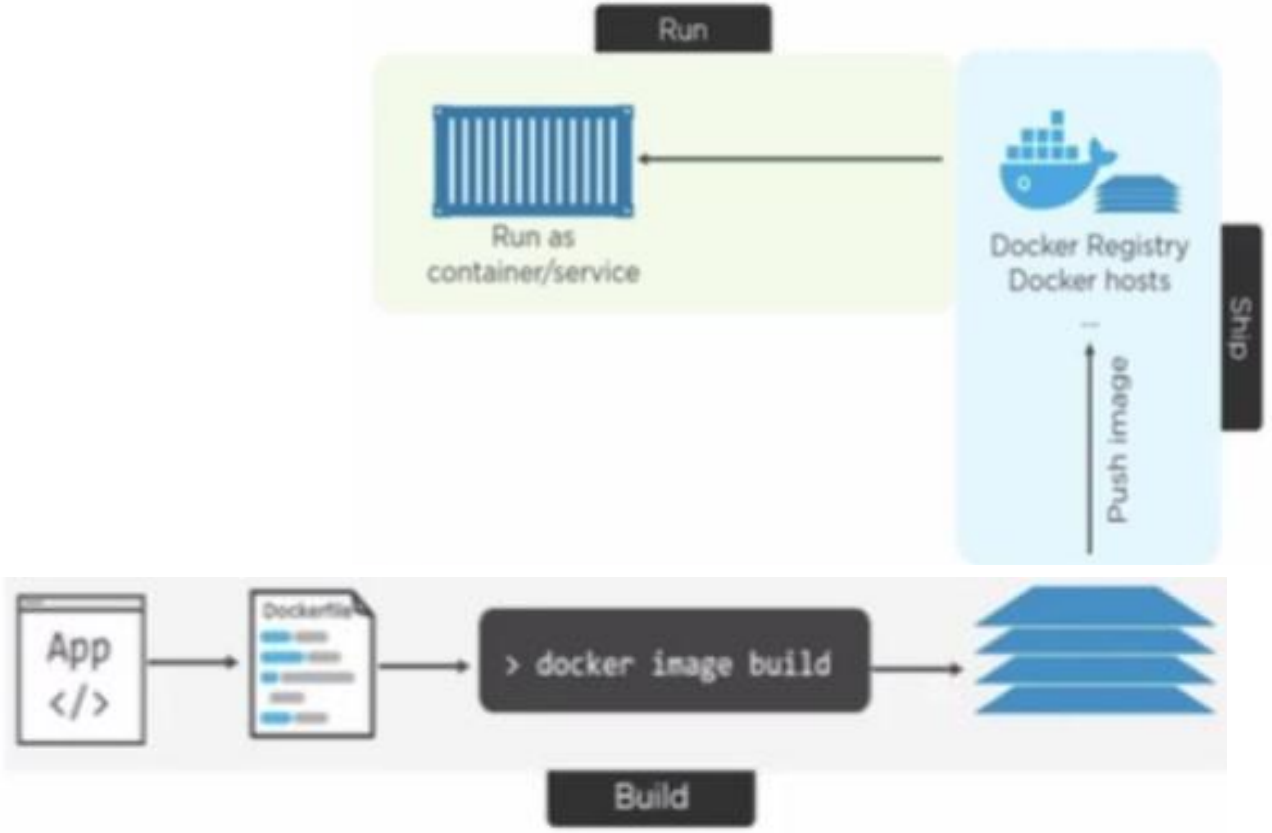


CONTAINERIZING AN APP FROM SCRATCH

The process of taking an application and configuring it to run as a container is called containerization or dockerization.

Steps:

- Start with your application code.
- Create a docker file that describes your app, its dependencies, and how to run it.
- Feed this docker file into the docker image build command to create an image.
- Create and run a container from that image.



DOCKER FILE

DOCKER FILES FOR CONTAINERIZING AN APP

Html css js Application

Docker File

```
FROM nginx:latest
COPY . /usr/share/nginx/html
```

PHP Application

Demo

BIND MOUNT

BIND MOUNT

Data Persistence

Data does not persist when container no longer exists.

Docker has two options for containers to store files in that host machine so that the files are persisted even after the container stops

- Volumes
- Bind Mounts

By using bind mount, a file or directory on the host machine mounted into a container.

BIND MOUNT

Command for Bind Mount:

docker container run -it --name=test-application -v /home/meermurtza/testing-folder:/container-testing-folder usama9876/first-docker-app sh

echo "Testing Bind Mount" > msg.txt
ls msg.txt
cat msg.txt
Testing Bind Mount

- Delete this container

- Now bind another container with this mounted folder on host.

HAVE A GOOD DAY!