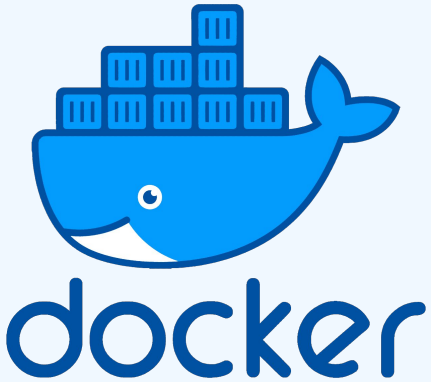
Introduction to Docker



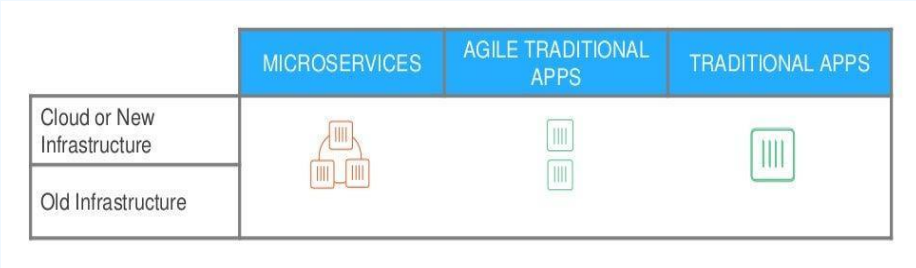
Transformation of Applications

Diverse Technology in Organisations

The Myth of Bi-Modal IT

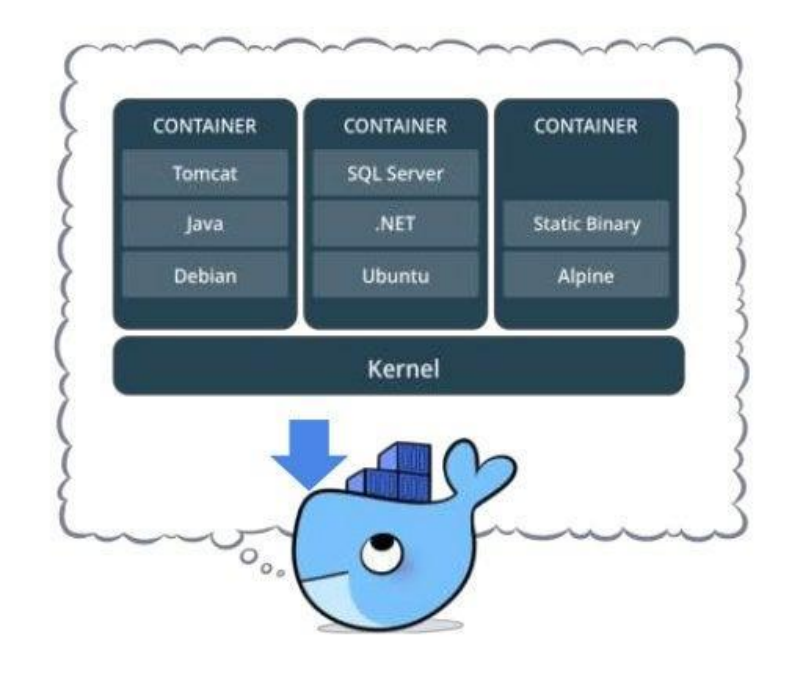
- Myth

- Reality



What is Docker ?

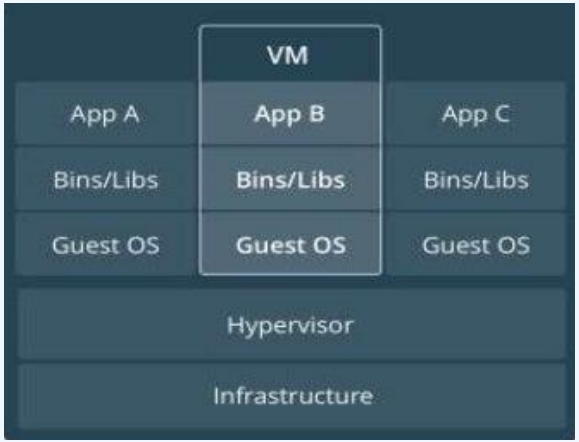
- It is a container technology.
- Standardized packaging for software and dependencies.
- Isolate apps from each other.
- Share the same OS Kernel.
- Works with all major Linux and Windows Server.
- Released on March 2013 as open source.



Docker Growth

Docker Containers vs Virtual Machine

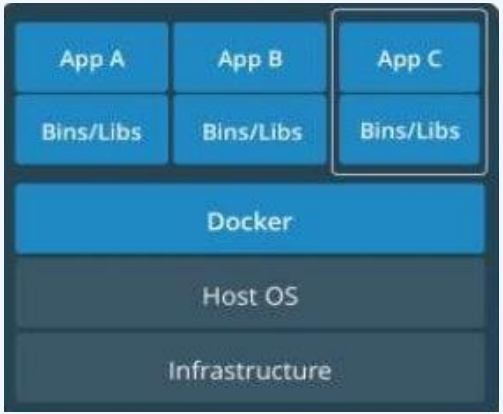
Containers are an app level construct



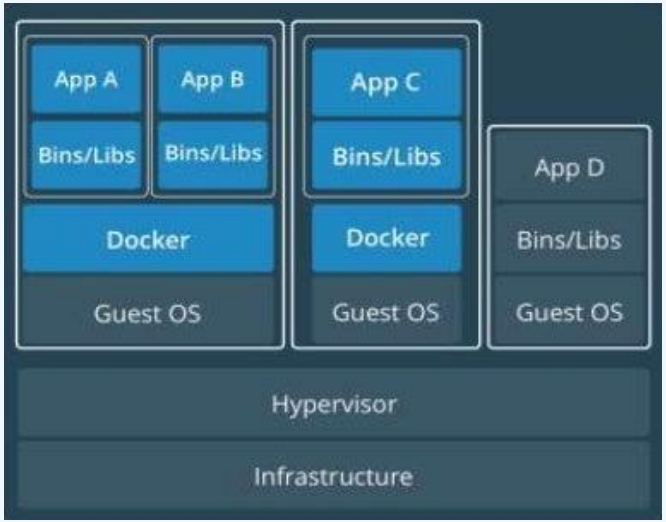
VMs are an infrastructure level construct to turn one machine into many servers

Docker Containers with Virtual Machine

DEV

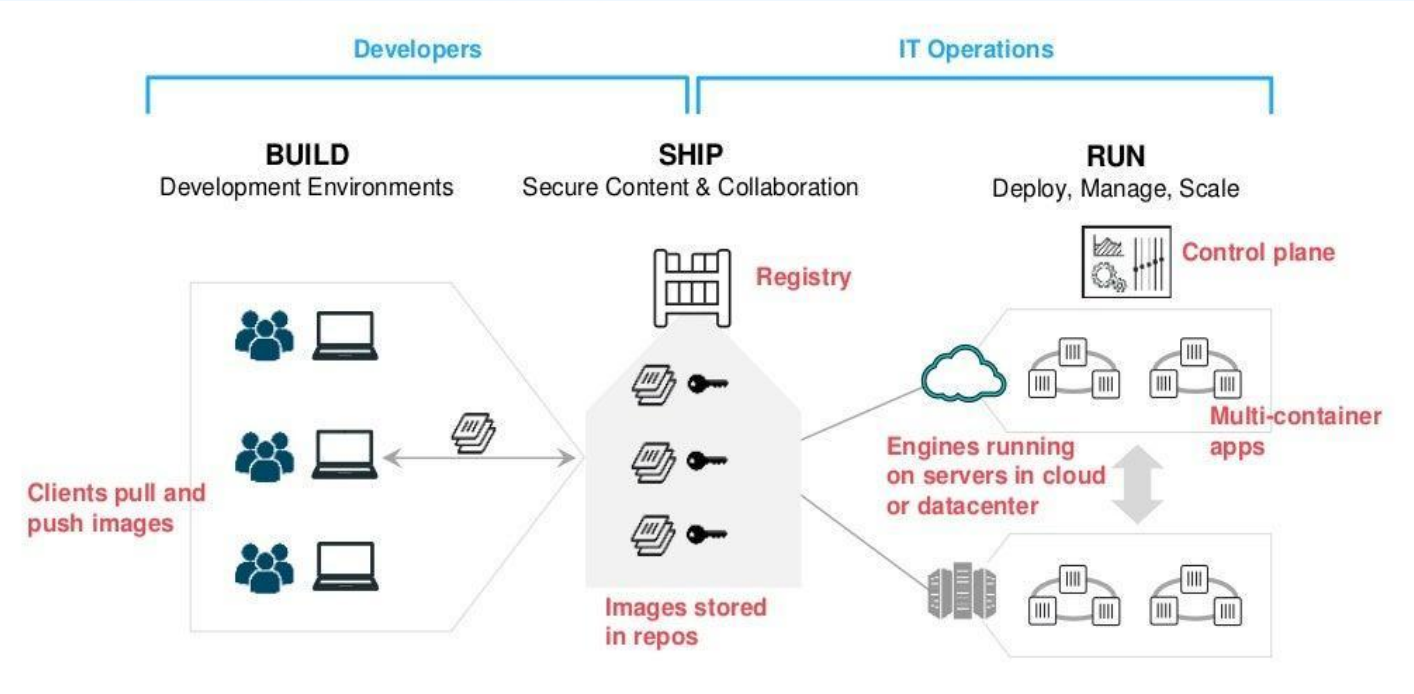


PROD



Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

Containers as a Service



Docker Basics

Docker Image

The basis of a Docker container. Represents a full application

Docker Container

The standard unit in which the application service resides and executes

Docker Engine

Creates, ships and runs Docker containers deployable on a physical or virtual, host locally, in a datacenter or cloud service provider

Registry Service (Docker Hub or Docker Trusted Registry)

Cloud or server based storage and distribution service for your images

Key Benefits of Docker Containers

Speed

- No OS to boot means, applications online in seconds.

Portability

- Less dependencies between process layers means, ability to move between infrastructure.

Efficiency

- Less OS overhead.
- Improved VM density.

Prerequisite

- Create a project folder.
- Open the project folder in Vs Code.
- Create a folder called ‘public’ inside the project folder.
- Create a ‘index.html’ file inside the public folder.

Dockerfile

- It's a text file that contains instructions for building a Docker image.
- It's like a batch script.
- The first line states the base image to begin with and then follow the instructions to install required programs, copy files and so on, until you get the working environment you need.
- Using docker build we create a docker images.

```
FROM nginx:1.23.0-alpine

COPY /public /usr/share/nginx/html

CMD ["nginx", "-g", "daemon off;"]
```

Docker Image

- A package with all the dependencies and information needed to create a container.
- An image includes all the dependencies plus deployment and execution configuration to be used by a container runtime.
- An image form the container filesystem.
- An image is immutable once it has been created.
- Command `docker build .` is used to create a docker image from dockerfile.

```
docker build . -t test-server


docker run -it --rm -p 9000:80
test-server


docker run -it --rm -p 9000:80 -v
$(pwd)/public:/usr/share/nginx/html
test-server
```

Docker Compose

- Docker Compose is used for running multiple containers as a single service.
- Each of the containers here run in isolation but can interact with each other when required.
- Docker Compose files are very easy to write in YAML.
- Docker Compose is used to activate all the services (containers) using a single command.
- Multiple containers use single host.

```yaml
version: '3.7'

services:

  webserver:

    build:

      context: .

      dockerfile: docker/Dockerfile

    container_name: webserver

    ports:

      - 9000:80

    volumes:

      - ./public:/usr/share/nginx/html
```

Reference

- Docker 101 [click here]
- Docker Command [click here]
- Docker Hub [click here]

Thank You