# Deep dive into

"Docker 106"

By Shramik Awale

Docker

- **Solomon Hykes** who invented the Docker.
- Open source project started by **eponymous** company
- Started initially as **dotcloud**

Docker Container

- A set of platform as a service

- Use OS-level virtualization

- Isolated from one another

- Bundle their own software, libraries and configuration files

- Communicate with each other through well-defined channels.
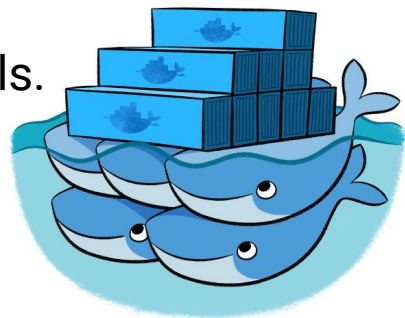
Features

Simplicity and Faster Configurations

Lightweight compare to virtualization

Run within host machine's kernel.

Run an application in a loosely isolated environment

Rapid Deployment

Alternatives for Hypervisor-based virtual machines

Docker Architecture

Based on client-server architecture.

Docker Client as a Client

- Communicate with more than one daemon

Docker Daemon as a Server

- Communicate with other daemons

Docker objects

Images

- Read-only template with instructions for creating a Docker container.

  e.g. Ubuntu with React js application and Nginx.

  docker pull ubuntu:latest

Docker objects

Containers

● Runnable instance of an image.

docker run -i -t ubuntu /bin/bash

Services

● Scale containers across multiple Docker daemons.
● Work together as a swarm with managers and workers.
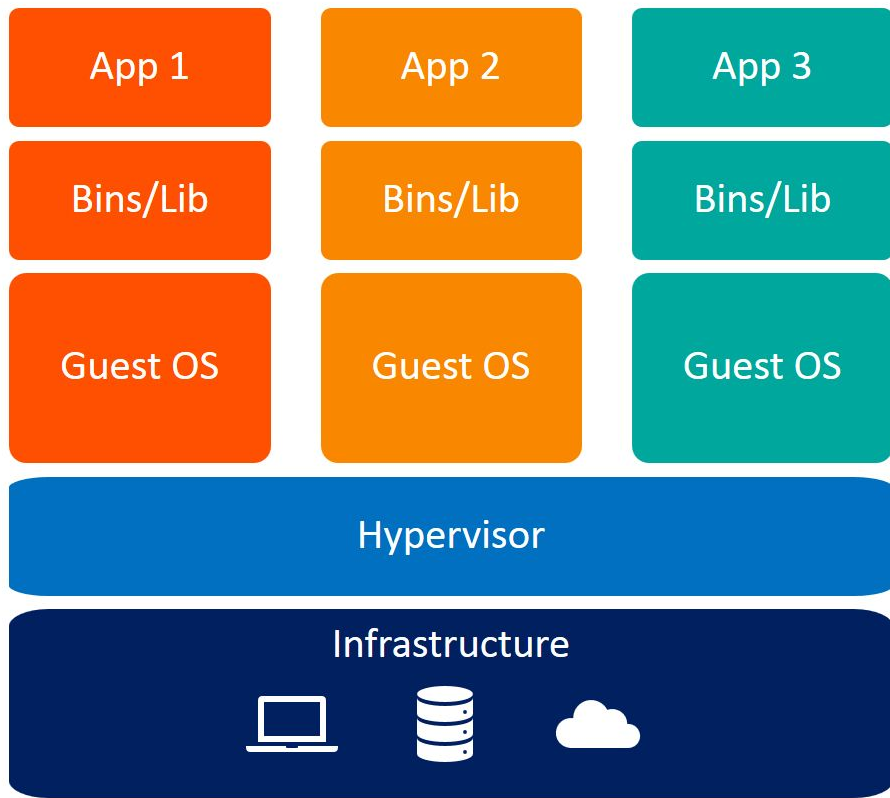
docker pull ubuntu:latest

Docker registries

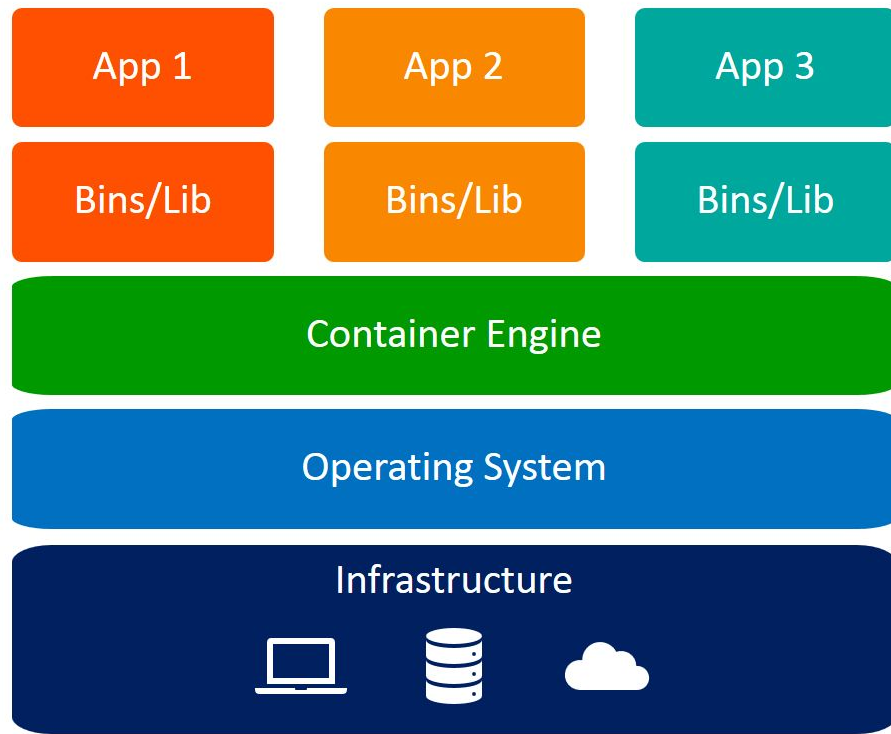Location to stores docker images publicly and privately.

Stores Docker images in **Docker hub**.

Underlying technology

- Namespaces : pid, net, ipc, mnt & uts.

- Control groups: enforce limits on available hardware resources.

- Union file systems: make them very lightweight & fast.

- Container format: Current format is libcontainer.

Virtual Machines — Containers

Virtualization	Docker
1. Heavyweight VMs are of few GBs	1. Containers are lightweight (KBs/MBs)
2. Hardware level process isolation	2. OS level process isolation
3. Each VM has a separate OS	3. Each container can share OS
4. Boots in minutes	4. Boots in seconds
5. Ready-made VMs are difficult to find	5. Pre-built docker containers are easily available
6. VMs can move to new host easily	6. Containers are destroyed and replicated rather than moving
7. Creating VM takes a relatively longer time	7. Containers can be created in seconds
8. More resource usage	8. Less resource usage
9. Fully isolated and hence more secure	9. Process-level isolation, possibly less secure

Let's try some docker image to build and run.
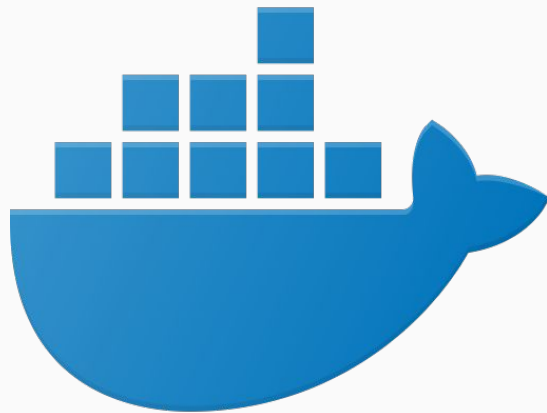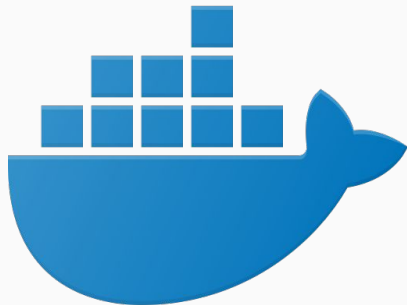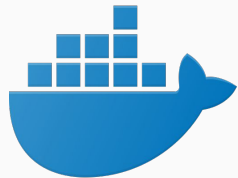
Pros and Cons

- Fast and use low resources
- Easy to deploy and extreme application portability
- Eliminating the "Works on My Machine" situation
- Compatible for DevOps and CI/CD

- Isolation
- Security
- Networking
- Slower than hypervisor

Alternatives

Virtualbox

LXC Linux Container

Wox

Rancher

Apache Mesos

Kubernetes

Questions ?

Thank you !