DOCKER

BASIC CONCEPTS

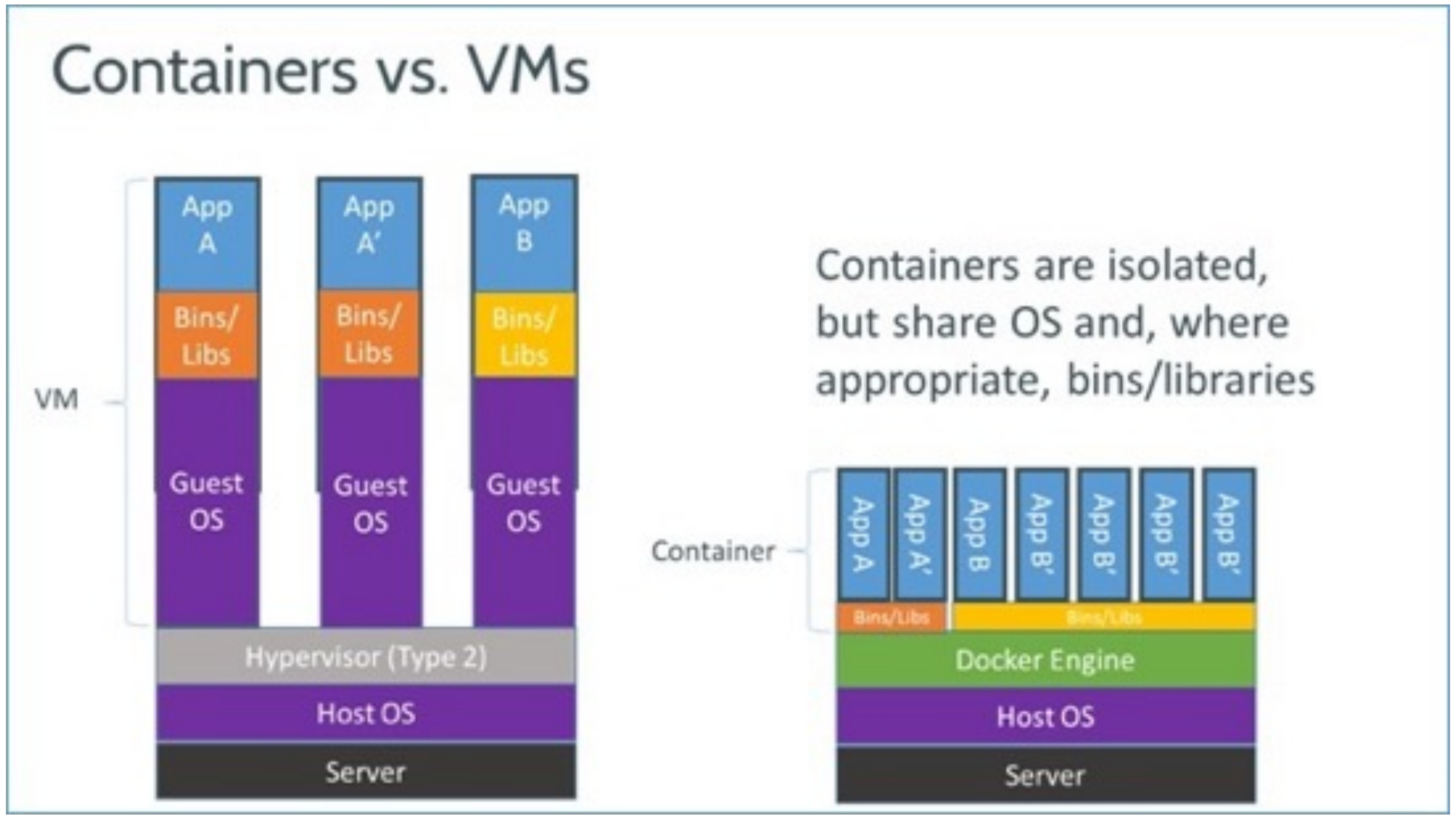
Intro

- An easy and lightweight way to model reality
  - Docker is fast. You can Dockerize your application in minutes. Docker relies on a copy-on-write model so that making changes to your application is also incredibly fast: only what you want to change gets changed.
  - You can then create containers running your applications.Most Docker containers take less than a second to launch.Removing the overhead of the hypervisor also means containers are highly performant and you can pack more of them into your hosts and make the best possible use of your resources.
- A logical segregation of duties
  - With Docker, Developers care about their applications running inside containers, and Operations cares about managing the containers. Docker is designed to enhance consistency by ensuring the environment in which your developers write code matches the environments into which your applications are deployed. This reduces the risk of ”worked in dev, now an ops problem.”
- Fast, efficient development life cycle
  - Docker aims to reduce the cycle time between code being written and code being tested, deployed, and used. It aims to make your applications portable, easy to build, and easy to collaborate on.
  - Encourages service orientated architecture

Introduction

- Docker is an open-source engine that automates the deployment of any application as a lightweight, portable, self-sufficient container that will run virtually anywhere.

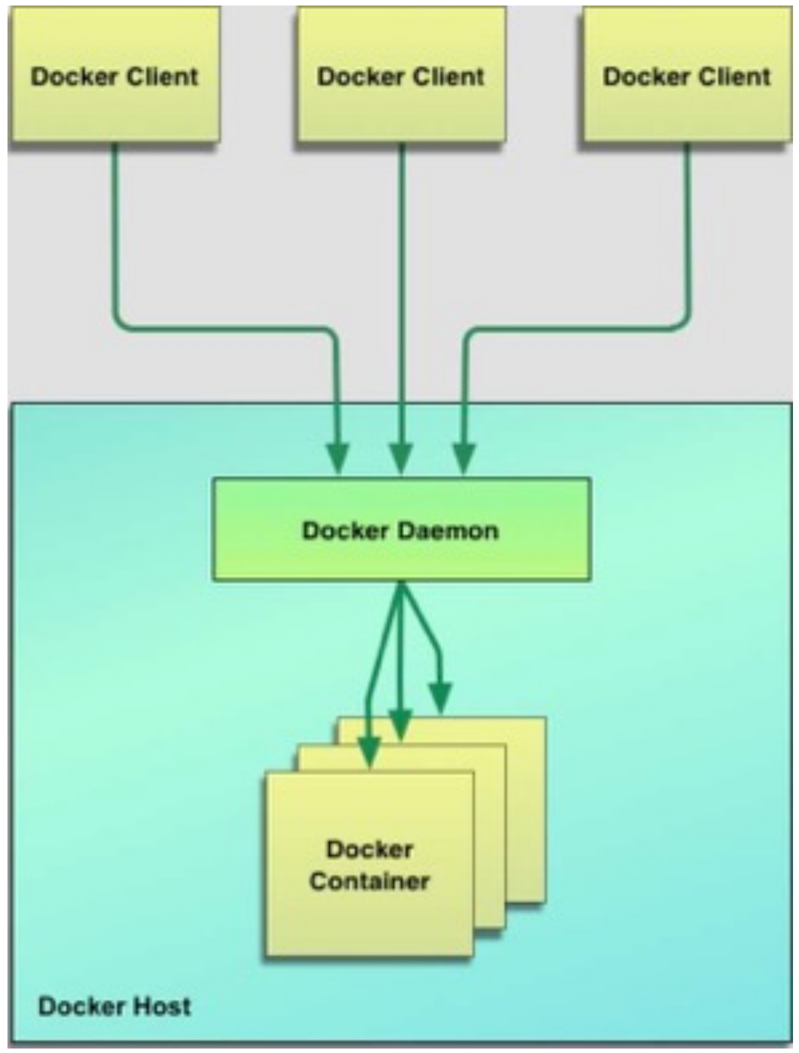
- Based on LXC (Linux Container), and AUFS (Union FS), easy to use.
- Similar to VM as end-user with different features



Docker components

- The Docker client and server
- Docker Images
- Registries
- Docker Containers

- Docker is a client-server application. The Docker client talks to the Docker server or daemon, which, in turn, does all the work. Docker ships with a command line client binary,docker,as well as a RESTfulAPI. You can run the Docker daemon and client on the same host or connect your local Docker client to a remote daemon running on another host.


Docker Client
Docker Client
Docker Client
Docker Daemon
Docker Container
Docker Host

Docker images

- Images are the building blocks of the Docker world. You launch your containers from images. Images are the ”build” part of Docker’s life cycle. They are a layered format, using Union file systems, that are built step-by-step using a series of instructions.

Registries

- Docker stores the images you build in registries. There are two types of registries: public and private. Docker, Inc., operates the public registry for images, called the Docker Hub.

Containers

- A Docker container is:
  - An image format.
  - A set of standard operations.
  - An execution environment.

- Think shipping container

What problems does Docker Solve

- Dev is in charge of what is in the container
- Ops is in charge of the container and where it runs, etc

- No more multi-page set of instructions to describe what is to be installed on a host to prep it for an application

- Couples very nicely with puppet/chef/vagrant, etc

THE DOCKER FILE