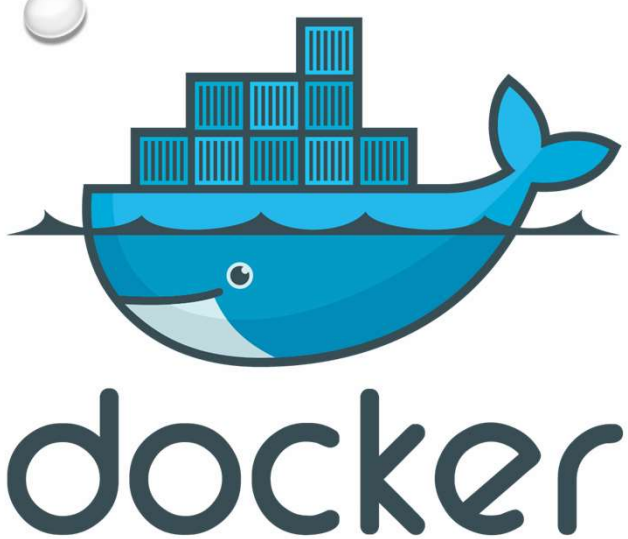


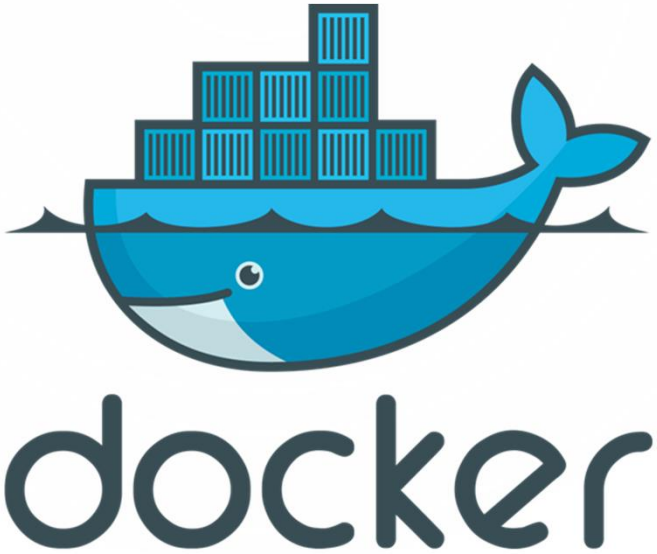
By Chidanand

Agenda

- Introduction – Participants & Trainer
- Setting the expectations
- Introduction to Docker, Docker vs VMs, Containers, Docker Engine, Registry, DockerHub
- Installation & Sample Docker exercises
- Docker command Line exercises
- Dockerizing Web Application, MySQL, Nginx
- Building your own Docker Image & Publishing it
- Q&A

Introductions

What is Docker?

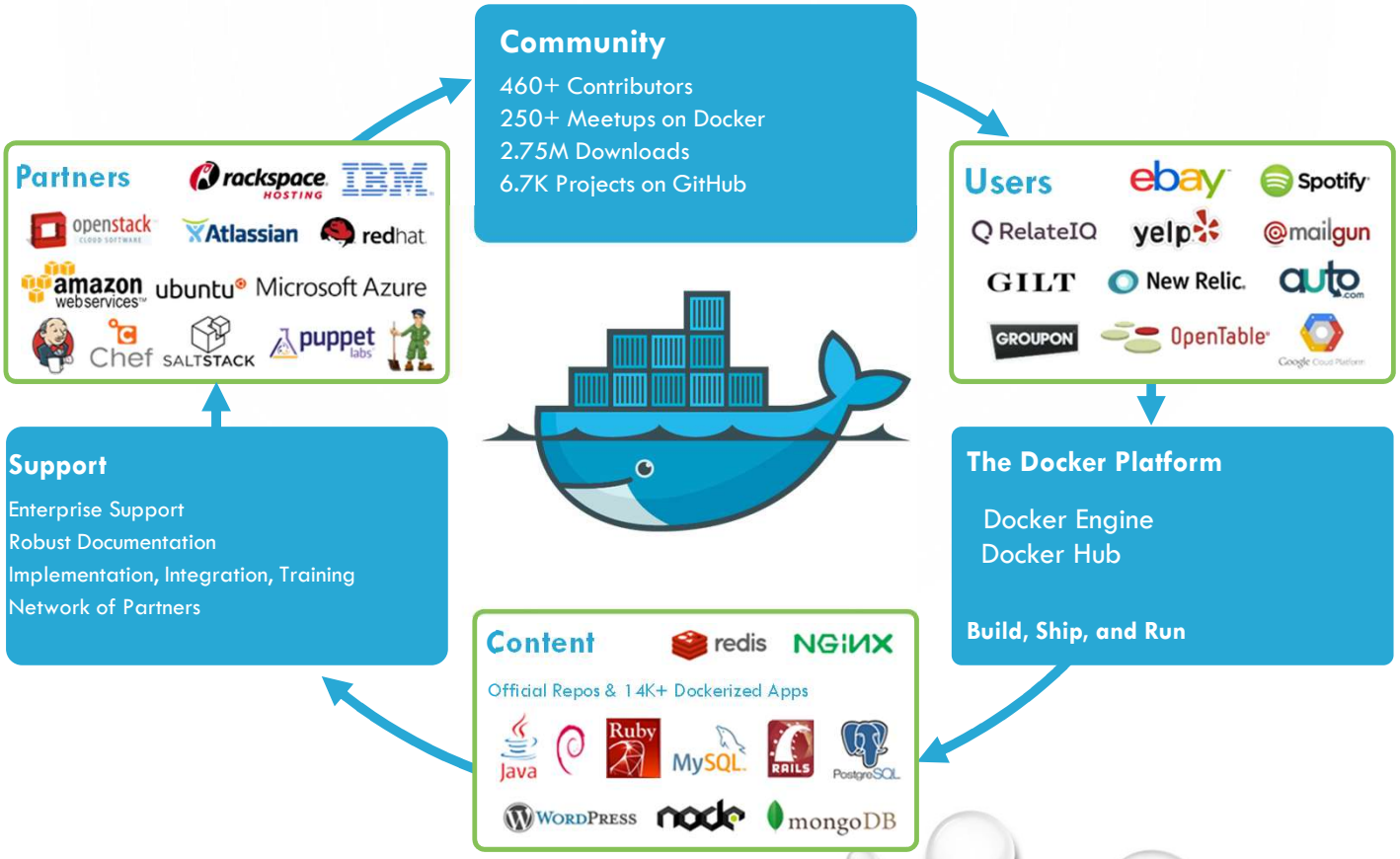


An Open Platform to Build, Ship, and Run Distributed Applications

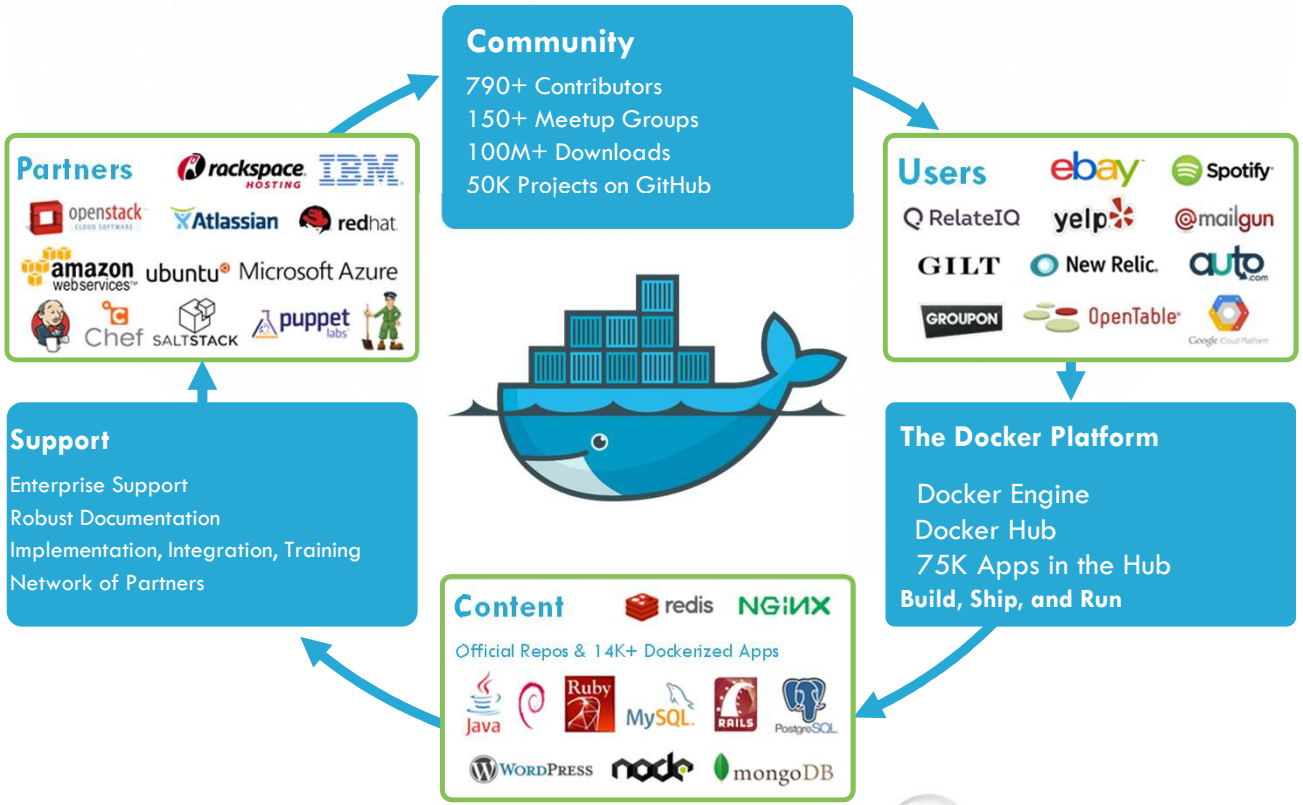
Docker

- Open Sourced in March 2013
- Wide Adoption in couple of years
- Huge Ecosystem was built around it
- Got the DevOps community super-excited

15 Months later – Incredible platform



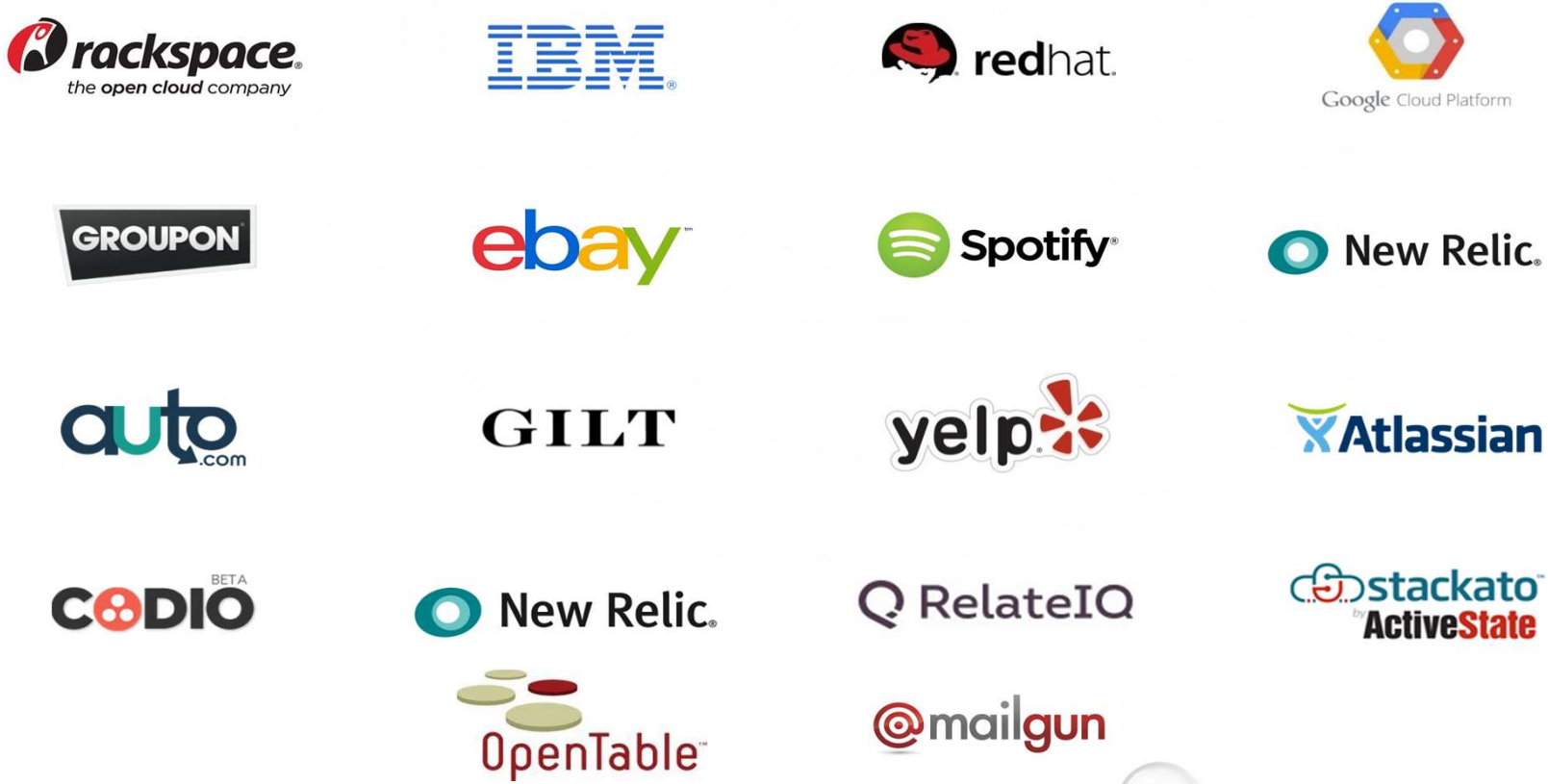
24 Months – Incredible platform & Ecosystem



Thanks to these Giants

- Namespaces (IBM)
- Cgroups (Google)
- LXC tools
- The Linux Kernel
- Git
- SELinux (Red Hat)
- Solaris Zones
- BSD Jails
- +++

Thanks to these users/use cases



Thank these Partner Ecosystem

Service Providers

rackspace, DEIS, Google Cloud Platform, amazon web services, IBM, SOFTLAYER, openstack, Yandex, OPENSHIFT, Microsoft Azure, DigitalOcean, CLOUD FOUNDRY, linode, Quay.io, Baidu 百度, Baremetal Industries, clever cloud, stackato, Google, domino, StackDock, Gilliam, tutum, ORCHARD, Flynn, VOXOZ

Operating Systems

redhat, boot2docker, CoreOS, ubuntu, debian, fedora, CentOS

Configuration Management

puppet labs, Chef, Jenkins, Capistrano, DotCi, Packer, ANSIBLE, SALTSTACK

Big Data

OpenCore, syncsort, hadoop

Dev Tools

Fig, VAGRANT, heroku, Shippable, runnable, NODECHECKER, wercker, drone.io

Service Discovery

NETFLIX Eureka, etcd, SkyDNS

Official Repositories

mongoDB, MySQL, PostgreSQL, Ruby, WordPress, Java, debian, CentOS, ubuntu, RAILS, redis, NGINX, node

Orchestration

MESOS, Shipper, CONSUL, VNS3, Marathon, fleet, ATOMIC

System Integrators

OpDemand, Wiredcraft, MomentumSI, InfoSiftr, Flux7, SHADOWSOFT, Dev9, relevance lab, IZURI

Thank You to the Ecosystem

Official Repositories

Dev Tools

Operating Systems

Big Data

Service Discovery

Build / Continuous Integration

Configuration Management

Infrastructure & Service Providers

Networking

Clustering & Scheduling

Storage

Management

Security

Monitoring & Logging

Consulting &Training

Docker definition

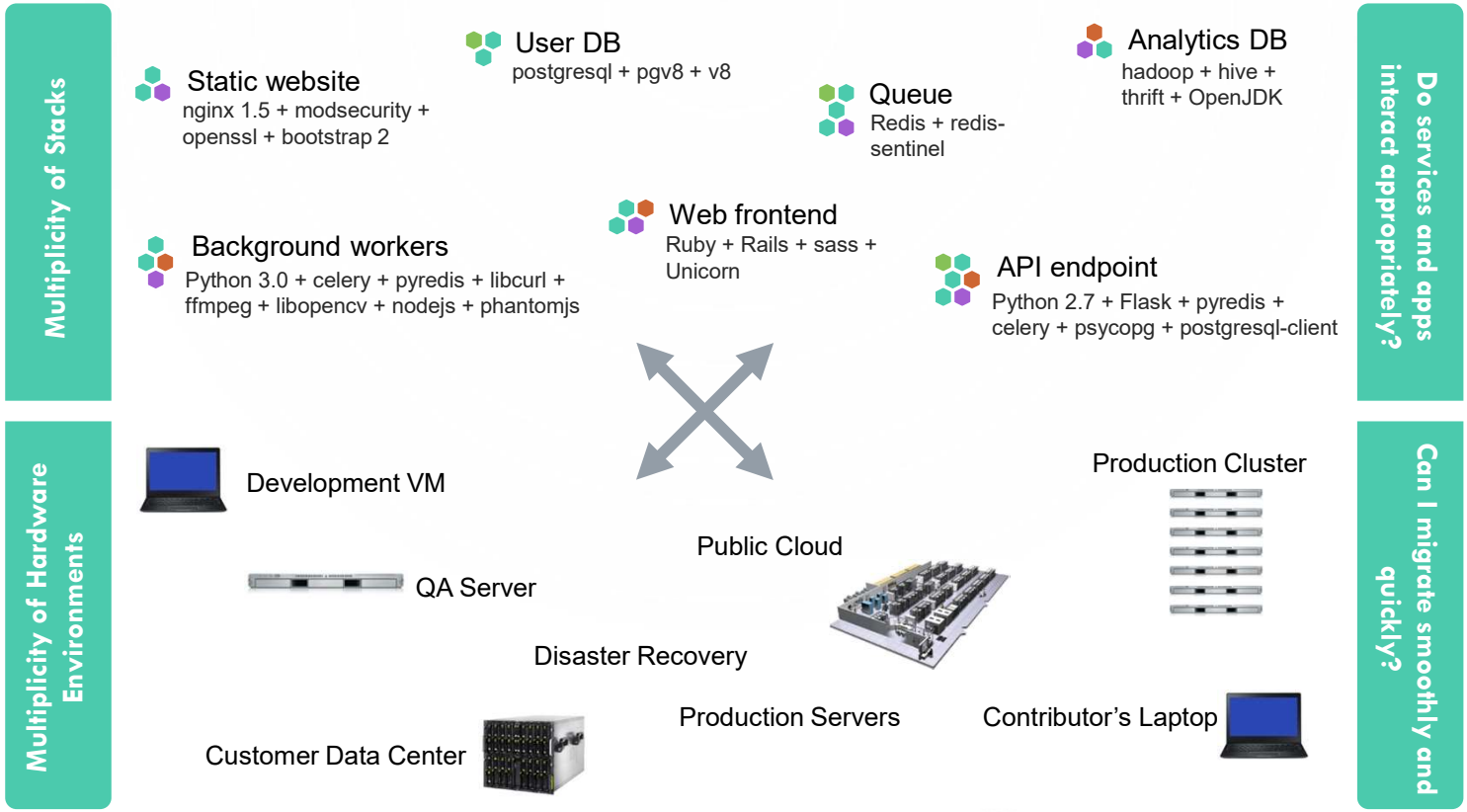
- Docker is an open source project that automates the deployment of applications within software containers
- An Open platform for developers & sys admins to build, ship and run distributed applications

…….But what's the big deal about this??

Application changes in the last few years

~2000	2014
Long lived	Development is iterative and constant
Monolithic and built on a single stack	Built from loosely coupled components
Deployed to a single server	Deployed to a multitude of servers

Application Problems



Resulting in NXN Compatibility matrix

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Related Anology

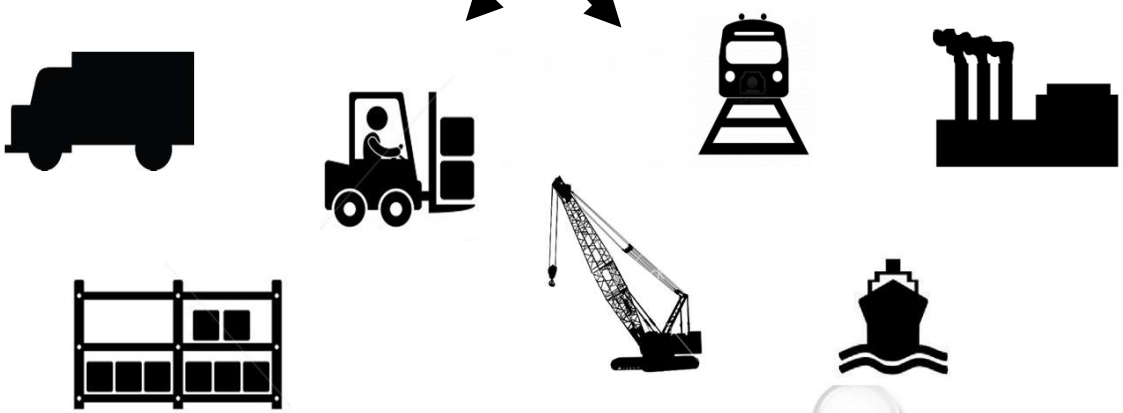
Cargo Transport Problems Pre-1960

Cargo Transport – Pre 1960

Multiplicity of Goods

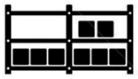
Do I worry about how goods interact (e.g. coffee beans next to spices)

Multiplicity of methods for transporting/storing



Can I transport quickly and smoothly (e.g. from boat to train to truck)

Also an NXN Matrix

	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

Solution – Intermodal Shipping Containers

Multiplicity of Goods

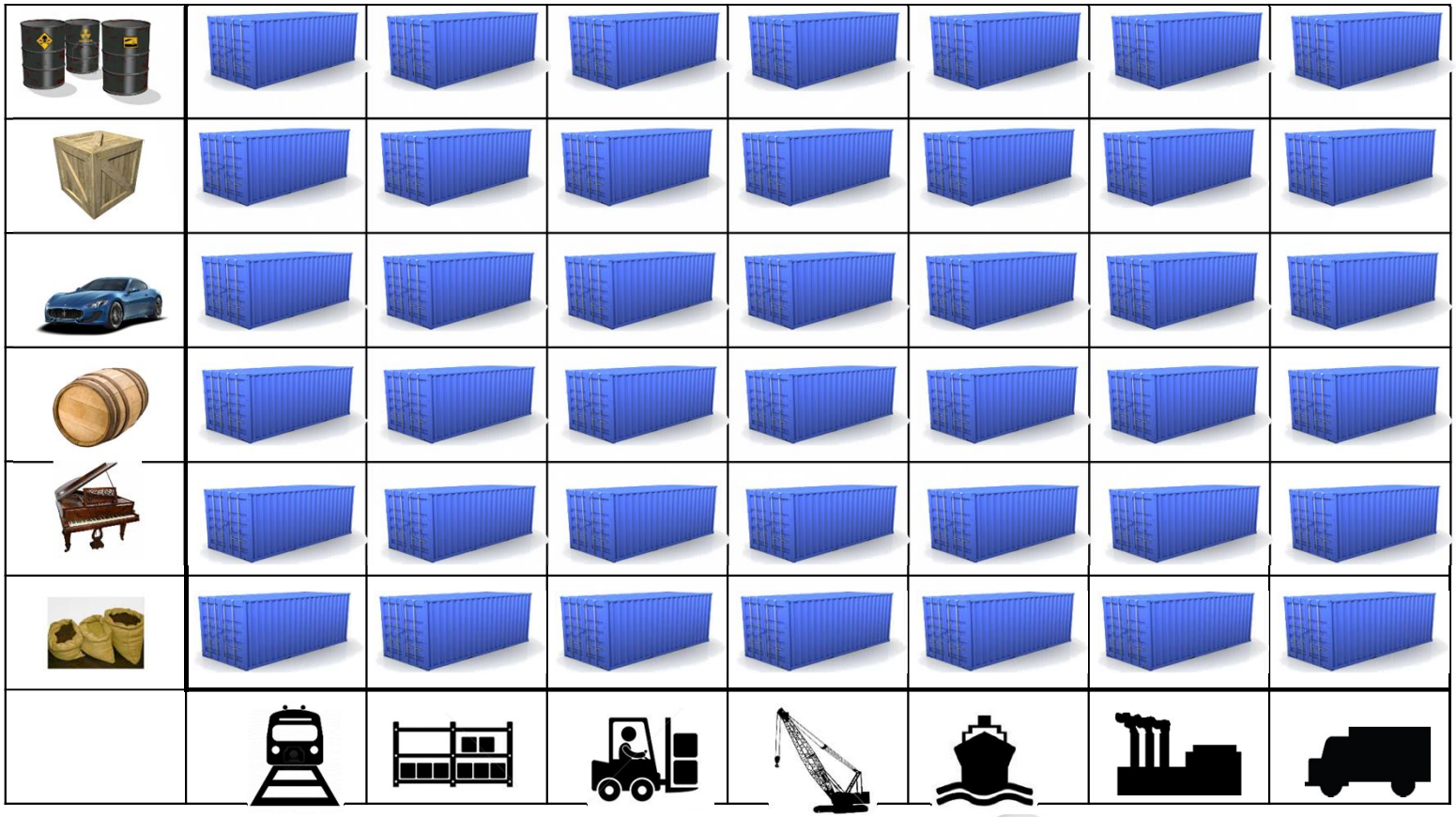
Do I worry about how goods interact (e.g. coffee beans next to spices)

A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

…in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another

Can I transport quickly and smoothly (e.g. from boat to train to truck)

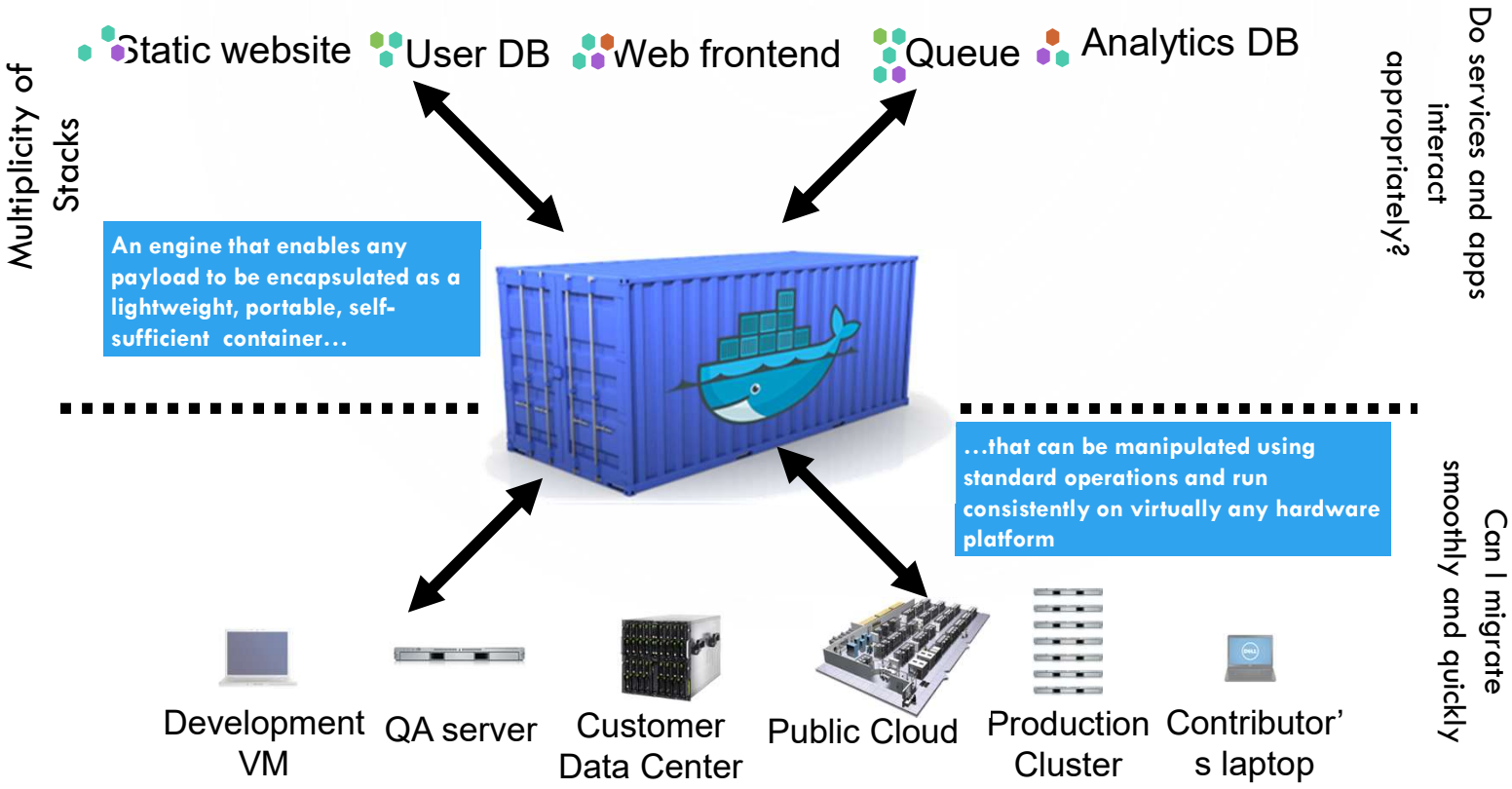
This eliminated the NXN Problem



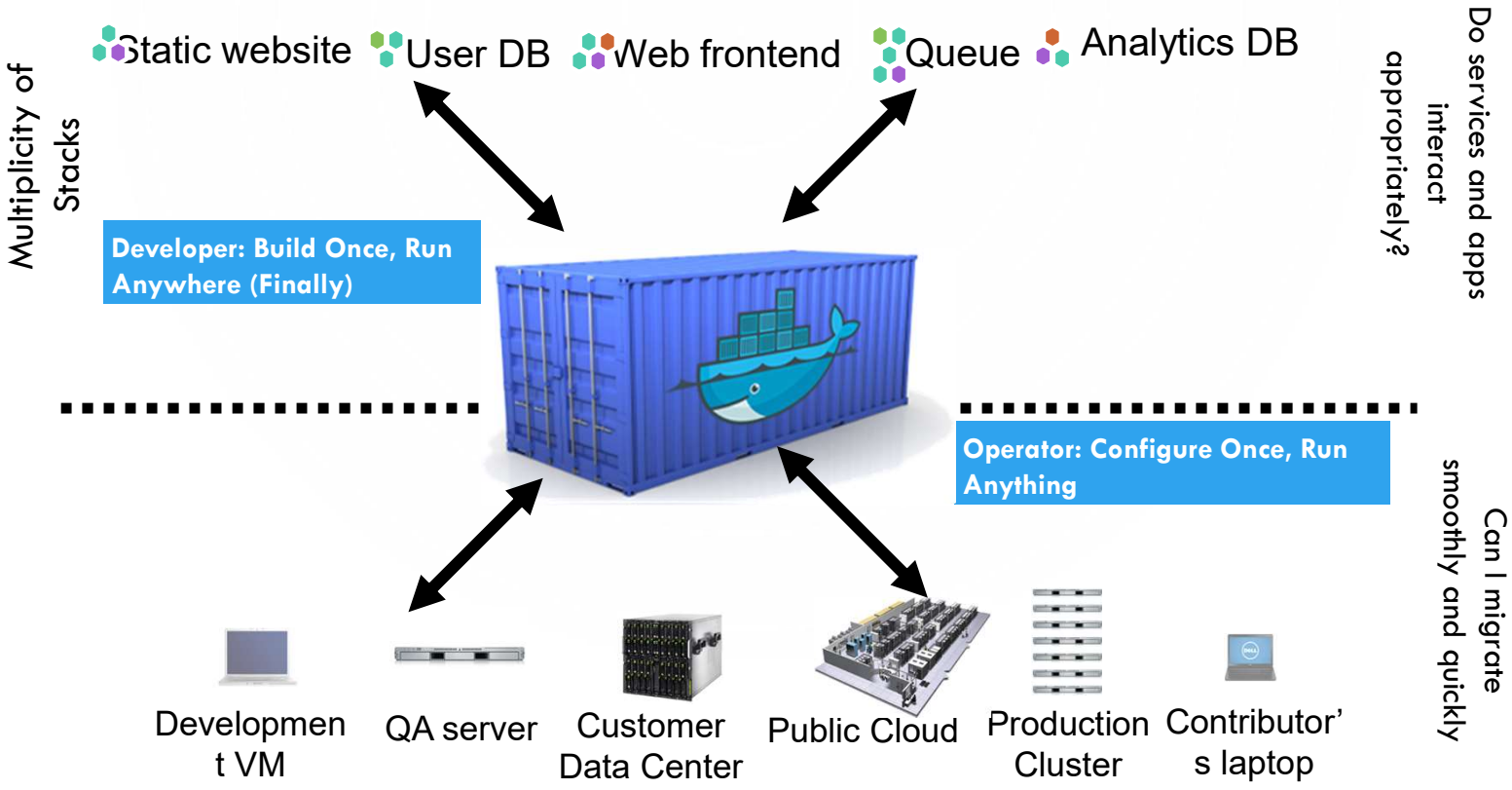
The Right Approach to containers matters

- Separation of concerns
- Automation
- Efficiency
- Broad ecosystem

Docker is a shipping container for Code



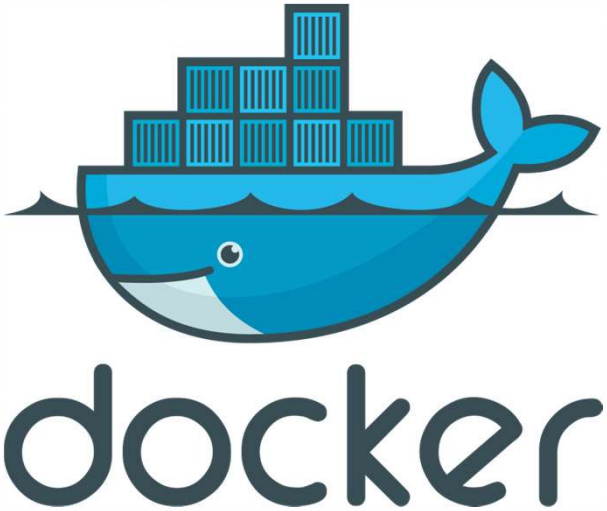
In short…..



Do services and apps interact appropriately?

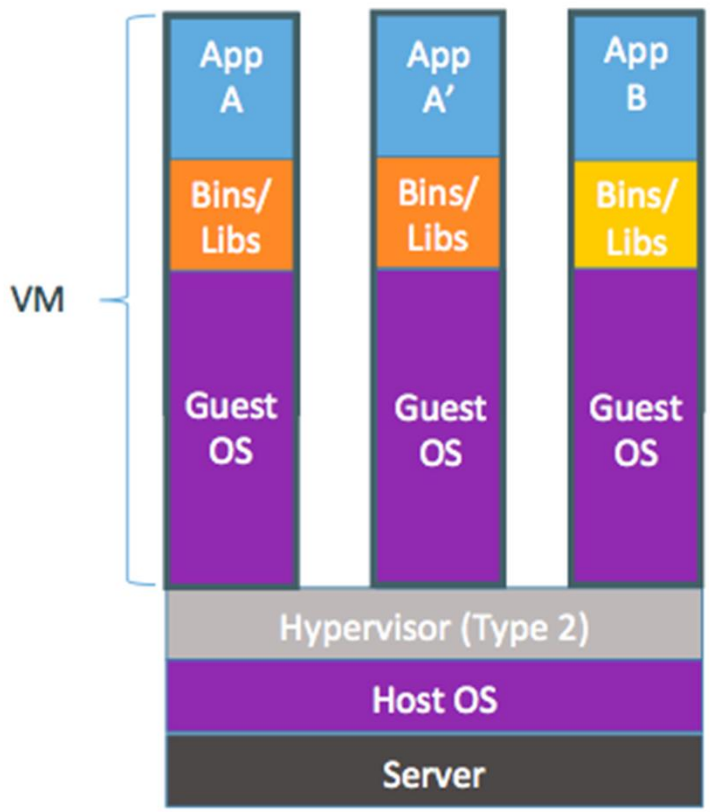
Can I migrate smoothly and quickly

Docker Definition

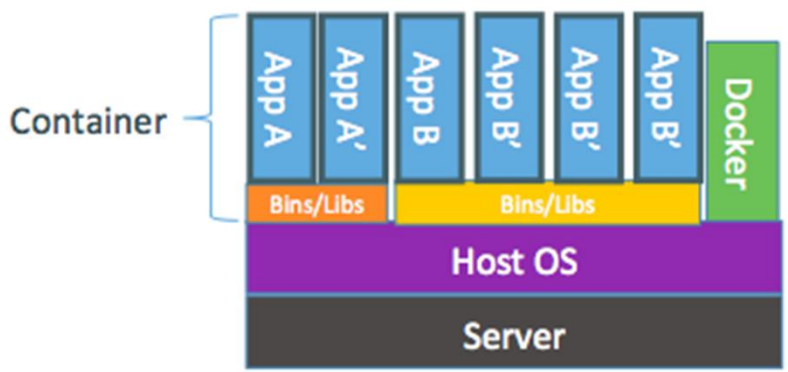


An Open Platform to Build, Ship, and Run Distributed Applications

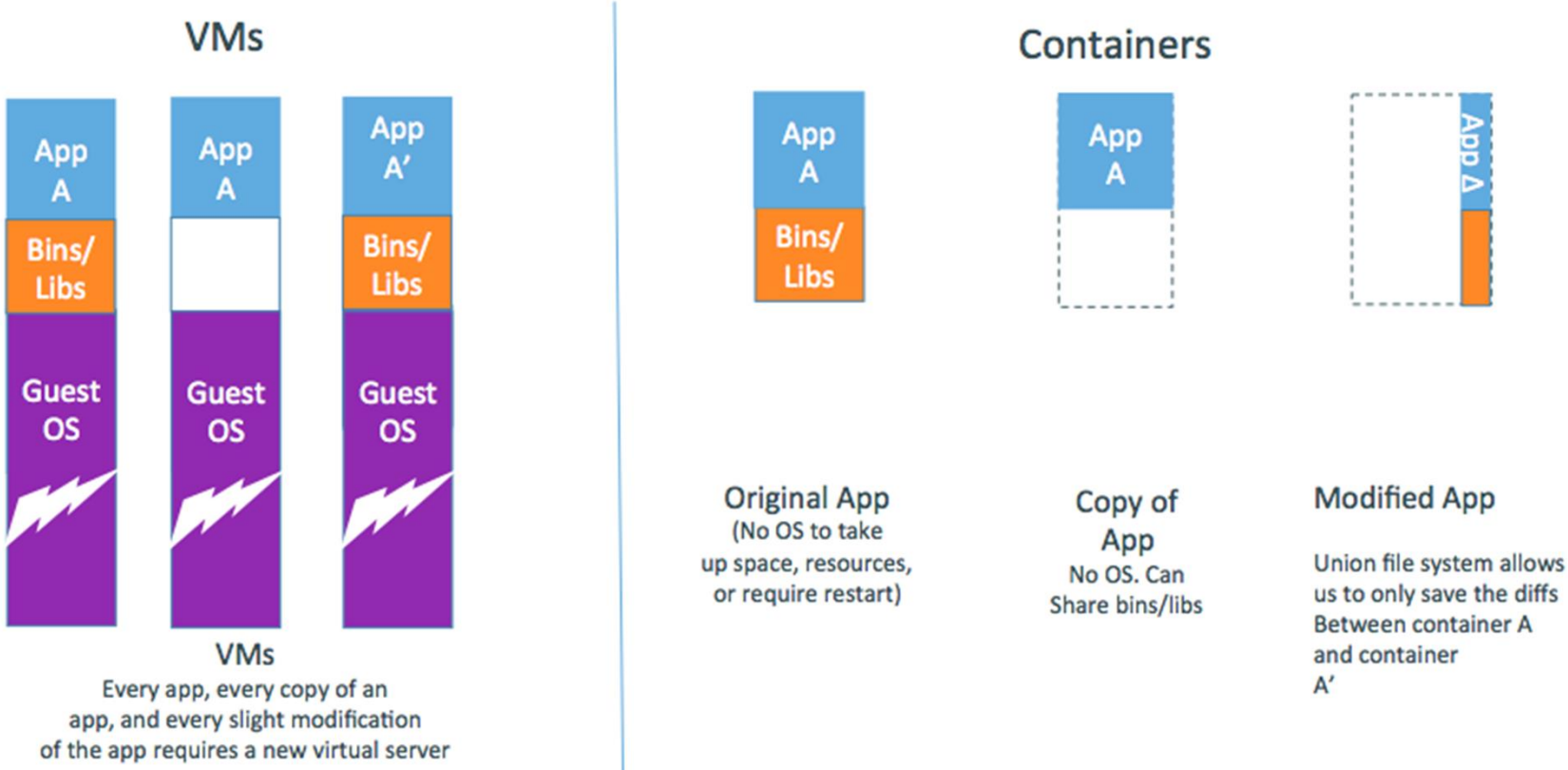
Container v/s VMs



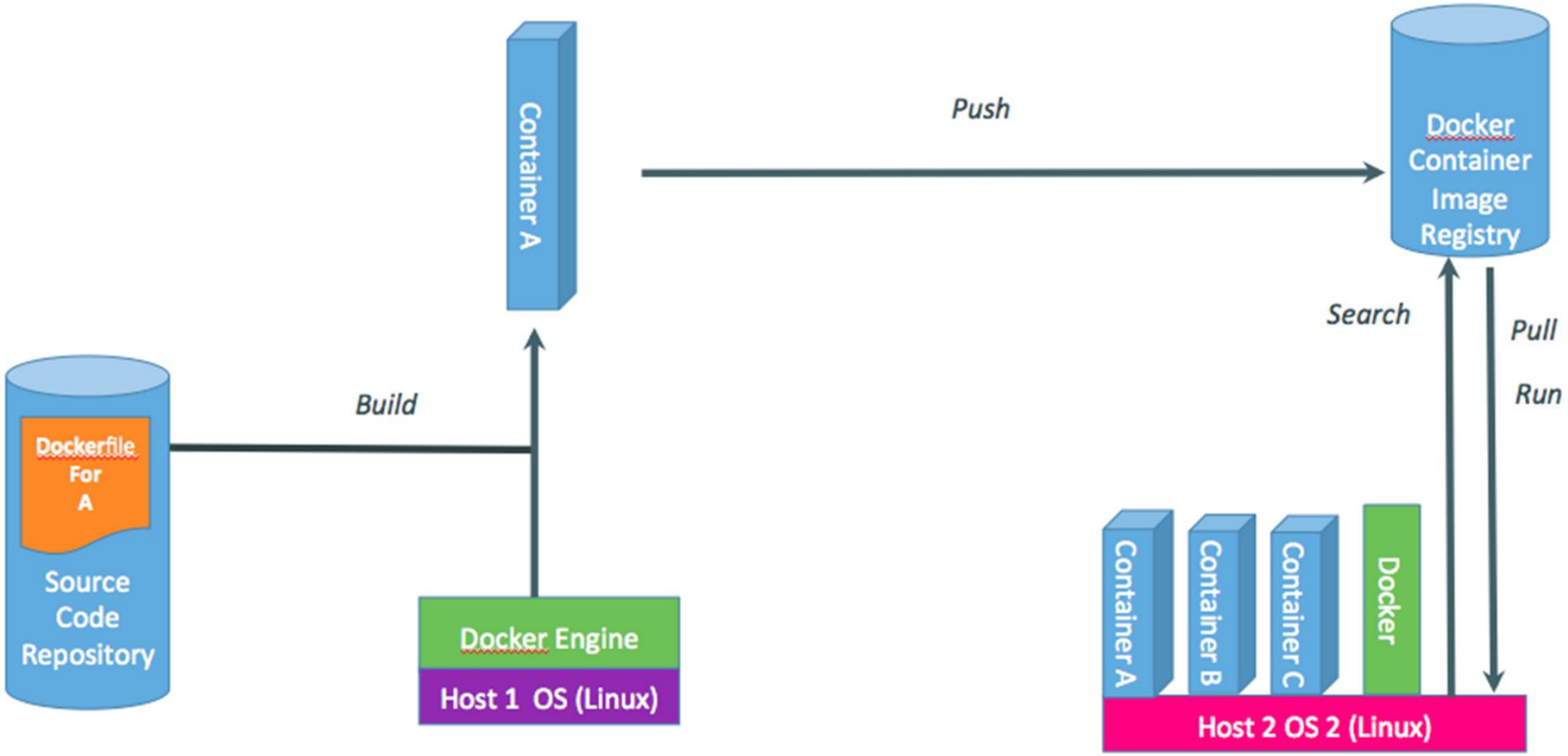
Containers are isolated, but share OS and, where appropriate, bins/libraries



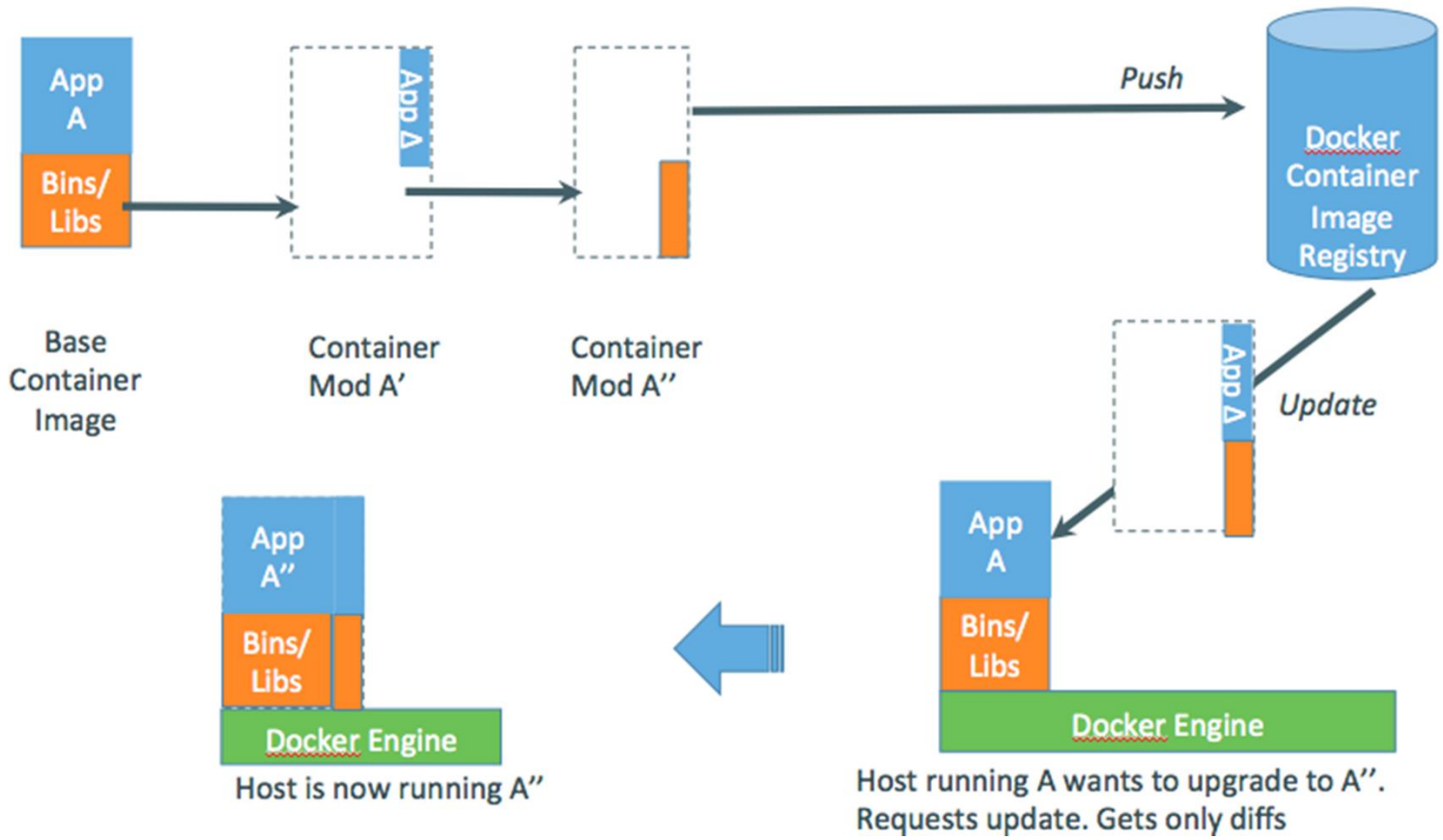
Why are Docker containers light weight



Docker Workflow



Changes & Updates



An Open Platform.....

Any App

+ 75K apps
+ 50K projects

API

Engine

open source software at the heart of the Docker platform

Hub

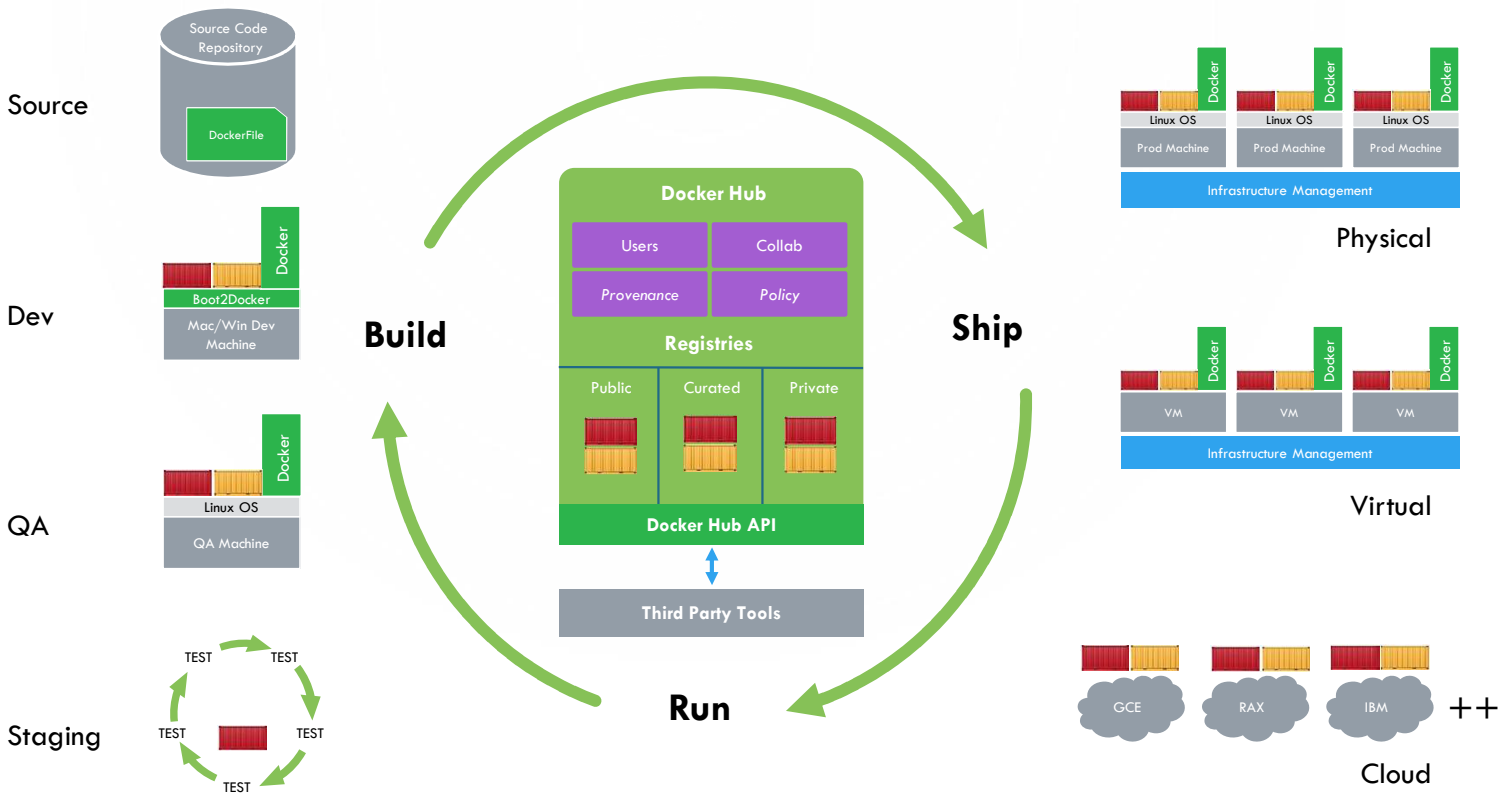
cloud-based platform services for distributed applications

API

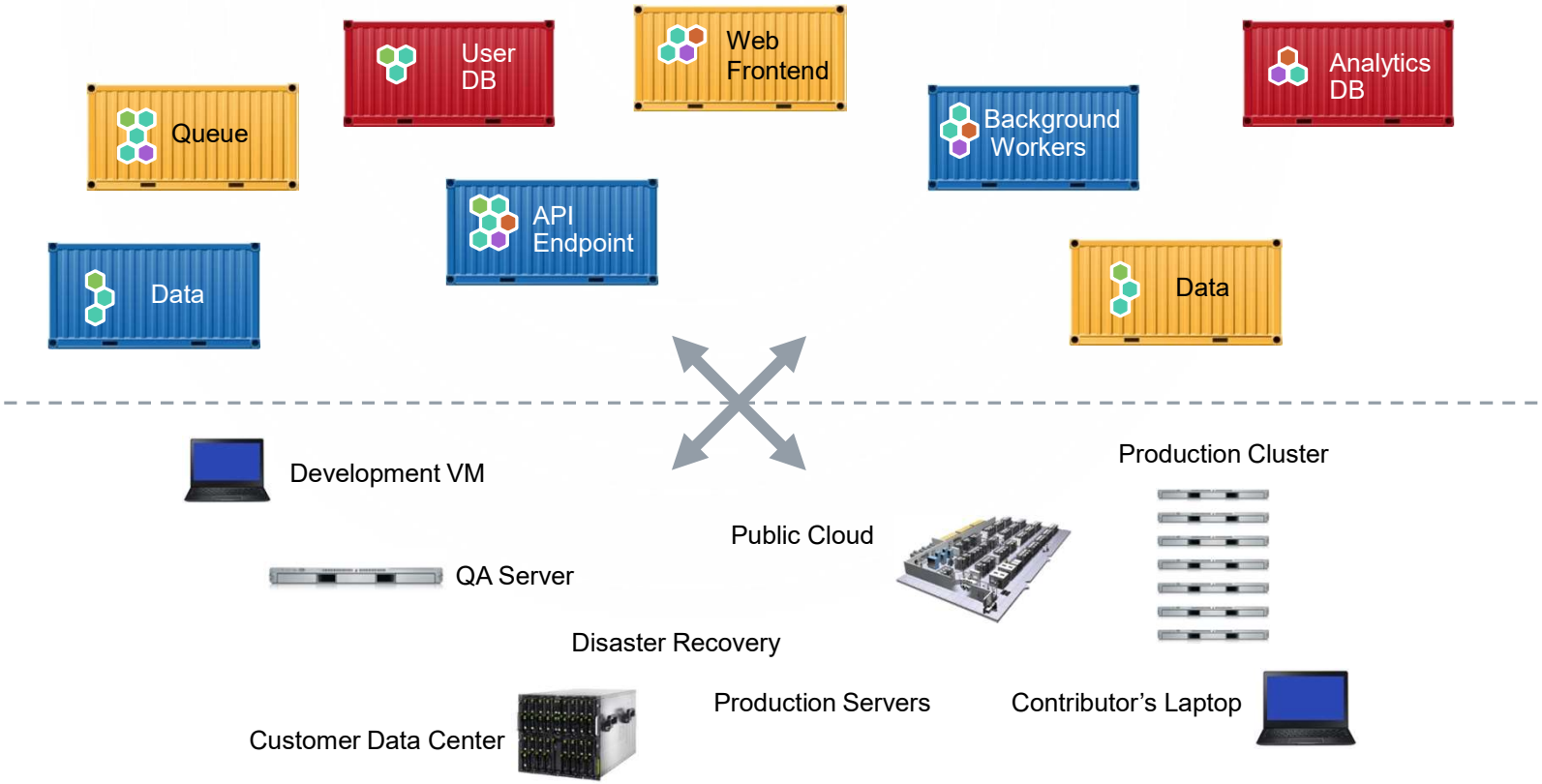
Any infrastructure

- Physical
- Virtual cloud

...to Build, Ship & Run



....Distributed Applications



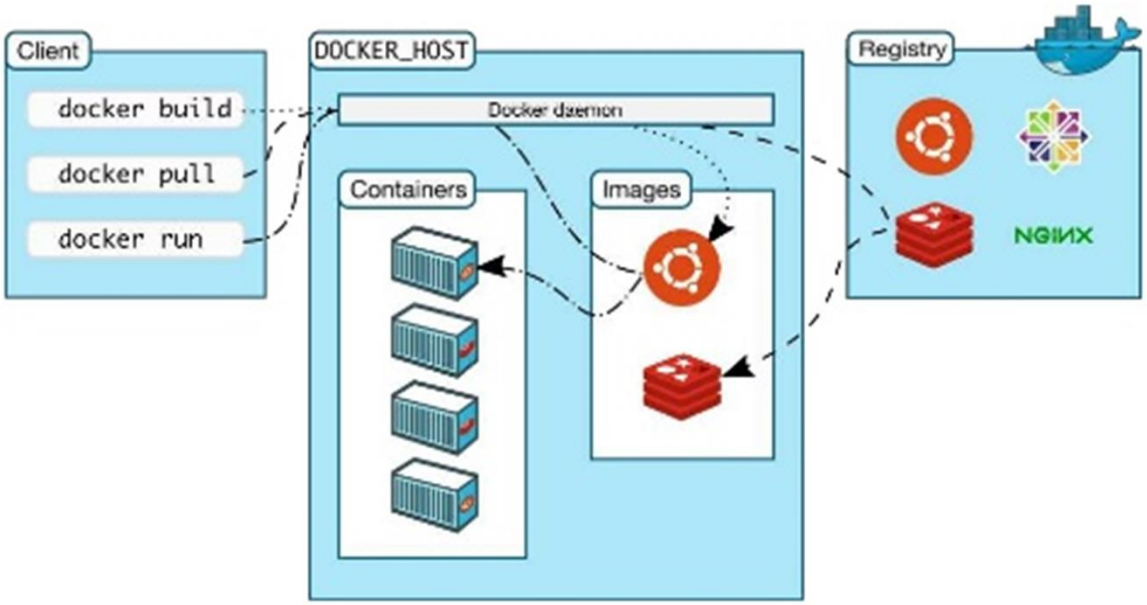
To wind up

Docker enables

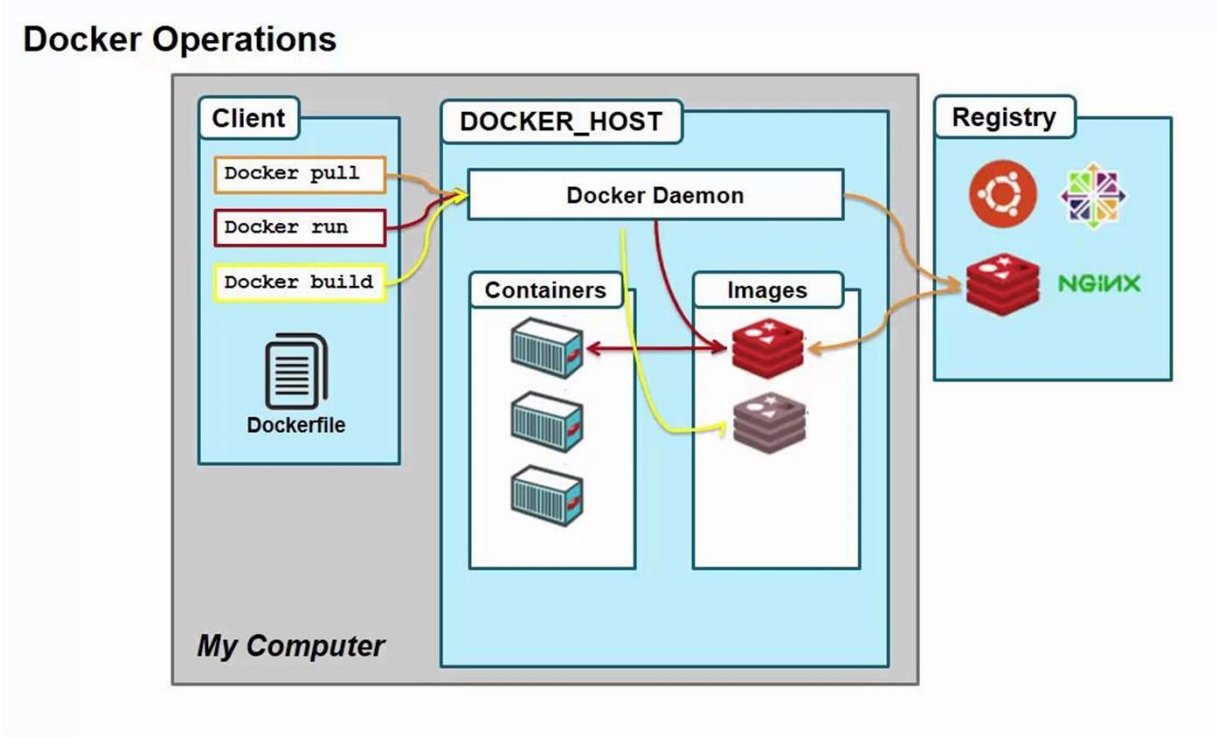
- Faster Delivery of your applications
- Deploying and scaling more easier
- Achieving higher density and running more work loads

Docker

Architecture



Docker Components & operations



Docker Vocabulary

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

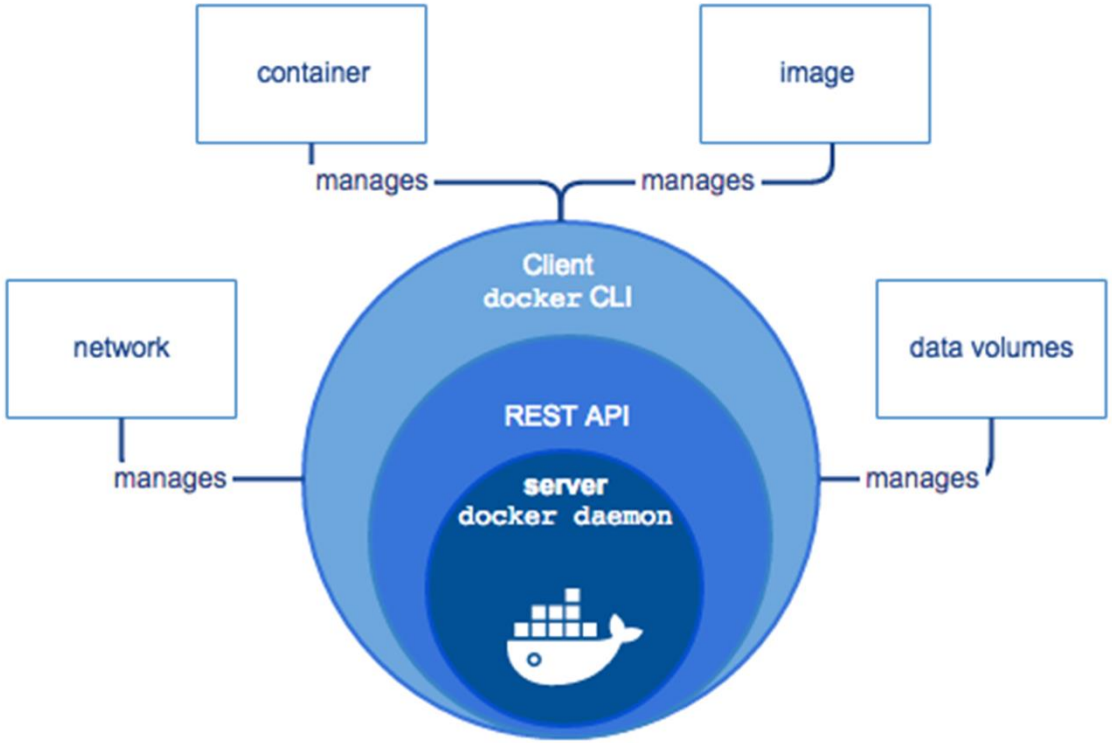
Docker Engine

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

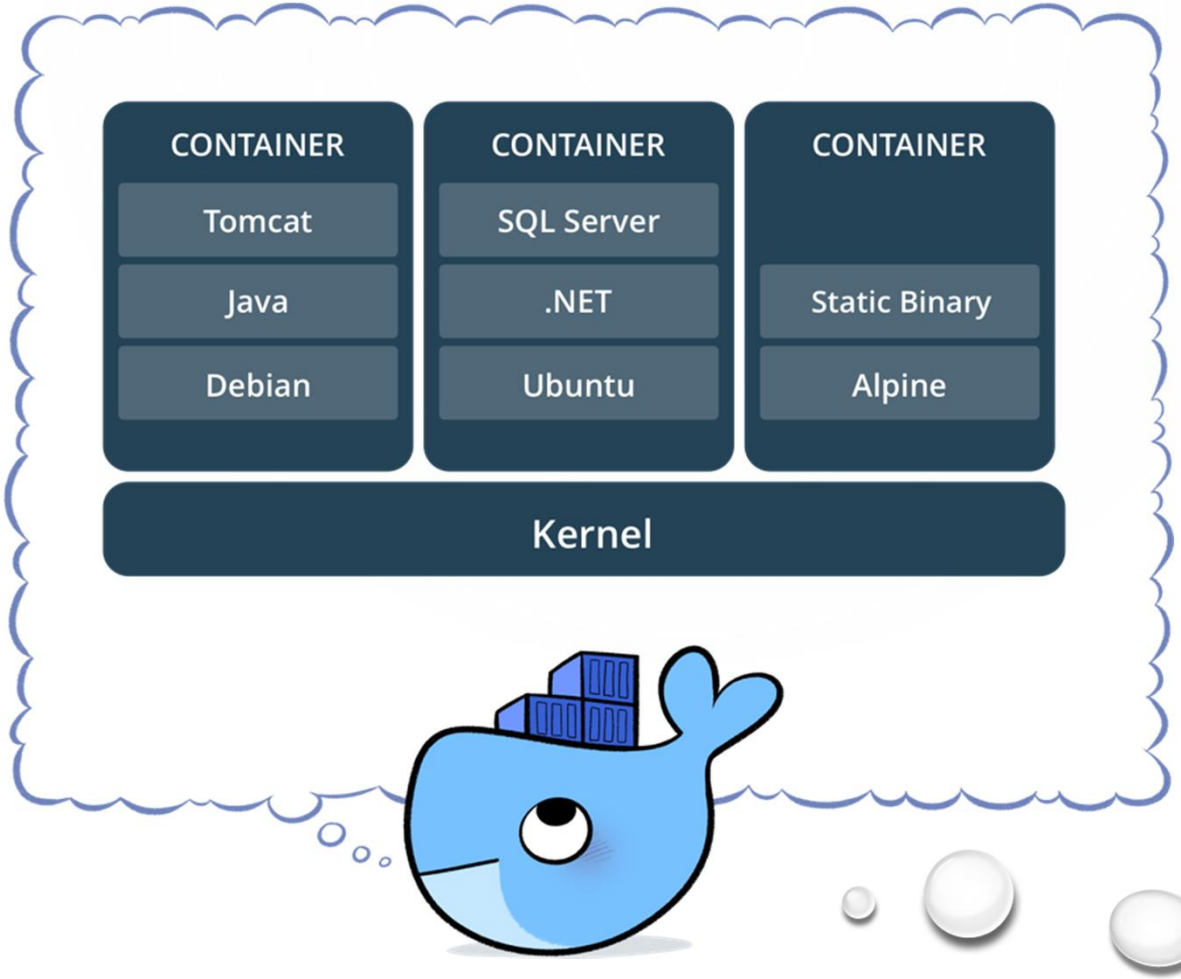
Registry Service (Docker Hub or Docker Trusted Registry)

Cloud or server based storage and distribution service for your images

Docker Engine



Docker Container



Docker Images

- Docker Image is a read only template Eg : Image could be a Ubuntu Operating System or an Ubuntu O/S with Apache, MySQL & a Web Application
- Images are used to create Docker containers
- New Image, update existing image or download images created by someone else & shared
- Docker images are the building components of Docker

Docker Installation

- Tons of popular platforms supported
- Windows 64, Mac OS, Linux (almost all flavours)
- Docker ToolBox : https://www.docker.com/products/docker-toolbox
- Install Docker Toolbox

Docker CLI Command Line Exercise

- $docker version
- $docker info
- $docker run –it hello-world

Docker Search for Images

- Docker Hub repository has tons of images, both official images & those contributed by other users
- https://hub.docker.com (Explore this portal)
- $docker search java
- $docker search tomcat
- $docker search mysql
- $docker search ubuntu

Docker get Ubuntu

- $docker pull ubuntu

OR

- $docker run –it ubuntu /bin/bash

This will pull latest alpine image from Docker hub and start it interactively

Docker run command

docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

$docker run –it ubuntu /bin/bash -> Will start Ubuntu in a container with a random name

$docker run –it --name My_Ubuntu ubuntu /bin/bash

Detaching from a container

CTRL+P, CTRL+Q -> Will detach from a container without stopping it

$docker attach CONTAINERID

Docker containers

- Containers will remain unless you explicitly remove them with $docker rm id/name
- To check running containers

$docker ps

- To check all containers

$docker ps -a

Housekeeping containers

- It is by design that stopped containers will exist so that they can restarted when required
- However, in case you want to automatically remove containers as soon as it exits, you can specify –-rm option when starting the container
- $docker run -it --rm ubuntu

Launching daemon containers

- Docker run –d option launches a container in a detached mode ie. Launch a daemon container
- $docker run -d ubuntu /bin/bash -c "while true; do date; sleep 5; done“
- $docker logs CONTAINERID
- $docker stop CONTAINERID

Inspect file system

- Docker Engine elegantly manages its file system and provides a diff subcommand to inspect changes in the container file system
- Create 3 files in a running container
- $docker diff CONTAINERID

Dockerfile

- Docker can build images automatically by reading instructions from a Dockerfile
- Dockerfile is a text document that contains all commands that a use can call on the command line to assemble an image

Dockerfile

- Create a Dockerfile using vi editor containing the following instructions

FROM alpine

CMD ["echo", "Hello World"]

Then build it using command

$docker build .

Dockerfile

- Successfully build IMAGEID
- $docker run –it –-name test IMAGEID

Dockerfile – Copy a script file

- Create a script.sh file in your local desktop and copy it within the Container image
- Use vi editor for creating the following script file

```
#! /bin/sh
echo hello world from script
```

$chmod 777 (give it executable permission

Dockerfile – Copy a script file

```
FROM alpine
COPY script.sh /script.sh
CMD ["/script.sh"]
```

```
$docker build .
$docker run –it –-name test IMAGEID
```

Questions

?