Introduction to Docker

Ajeet Singh Raina
Docker Captain – Docker, Inc.

docker

Who Am I?

- Sr. Systems Development Engineer at DellEMC
- 1st half of my career was in CGI & VMware
- 2nd half of my career has been in System Integration
- Testing/Project Lead for Dell EMC.
- Definitely more IT pro than developer
- @ajeetsraina (a frequent Twitterati)

The **{code} Catalyst** Program!

http://www.collabnix.com

Agenda

- A Shift from Monolithic to Microservices Architecture
- What is Docker? What problem does it solve for us?
- How Docker is different from VM & CM
- Using Docker: Build, Ship and Run WorkFlow
- Docker Ecosystem & Native Offering
- Docker Tools
- Demo

A Shift from Monolithic to Microservice Architecture

Applications have changed dramatically

Monolithic

Slow Changing

Big Server

A Decade Ago (and still valid)

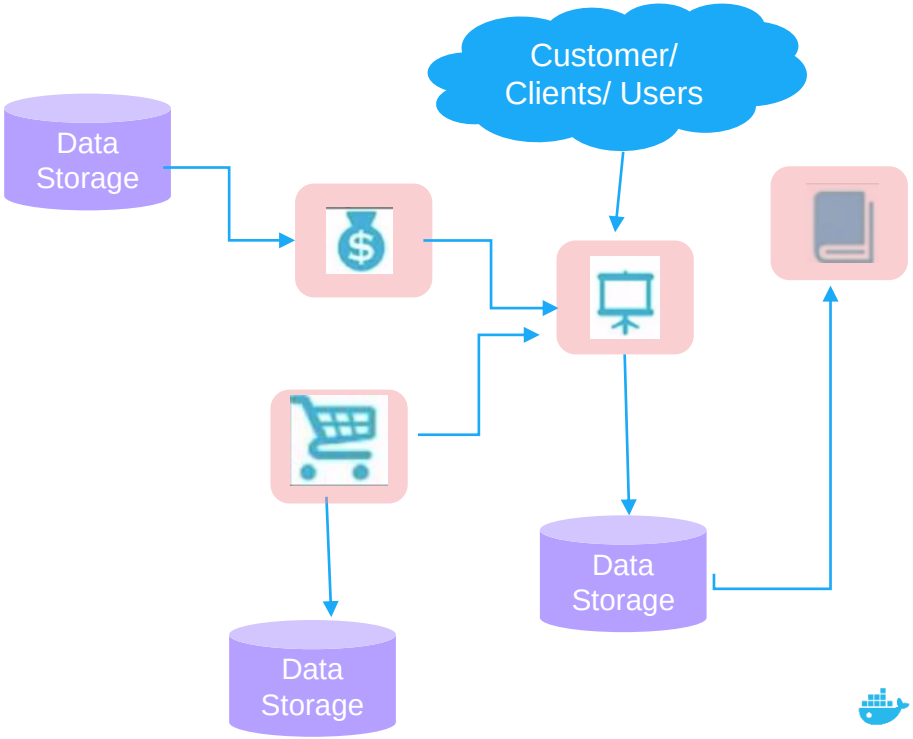
- Apps were monolithic
- Built on a single stack such as .NET or Java
- Long Lived
- Deployed to a single server

docker

Applications have changed dramatically

Today

- Apps are constantly developed
- Newer version are deployed often
- Built from loosely coupled components
- Deployed to a multitude of servers



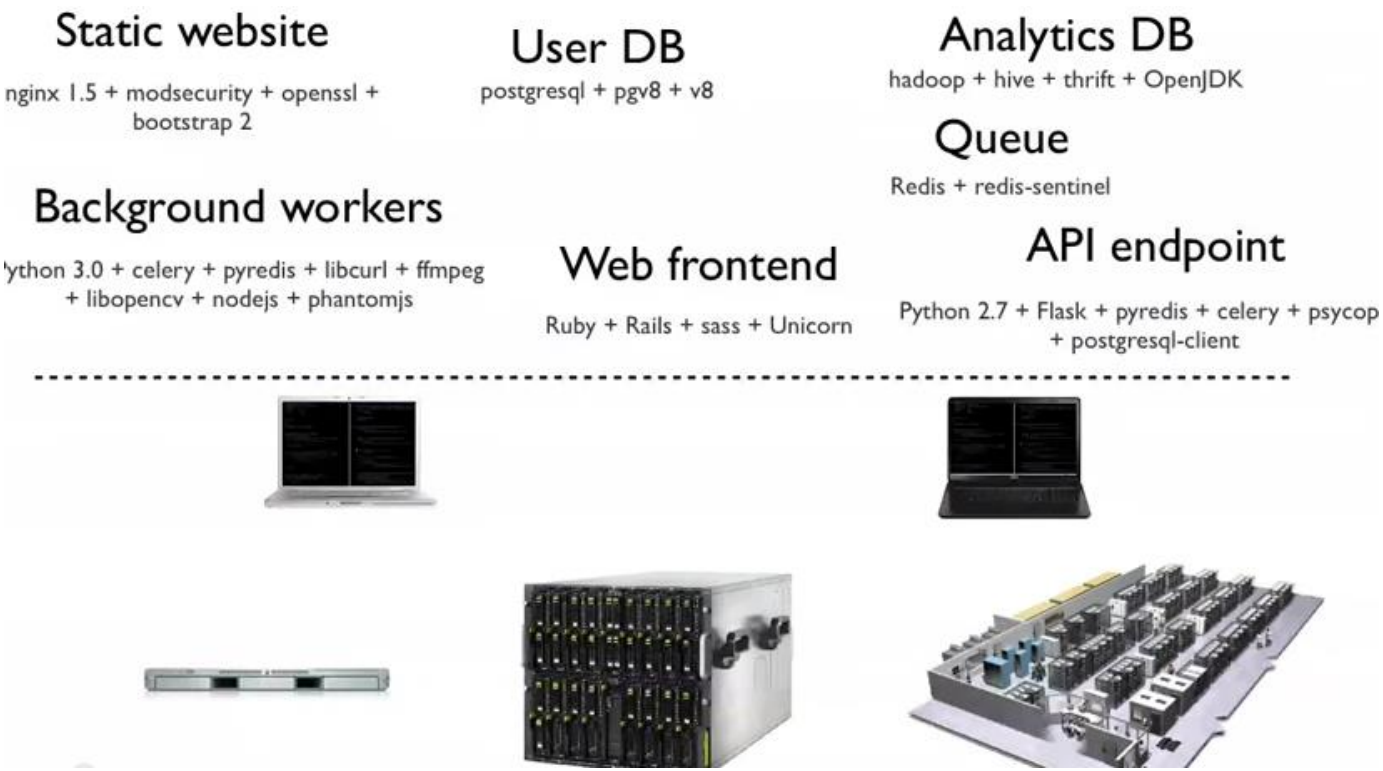
docker

Once upon a time… A Software Stack

LAMP

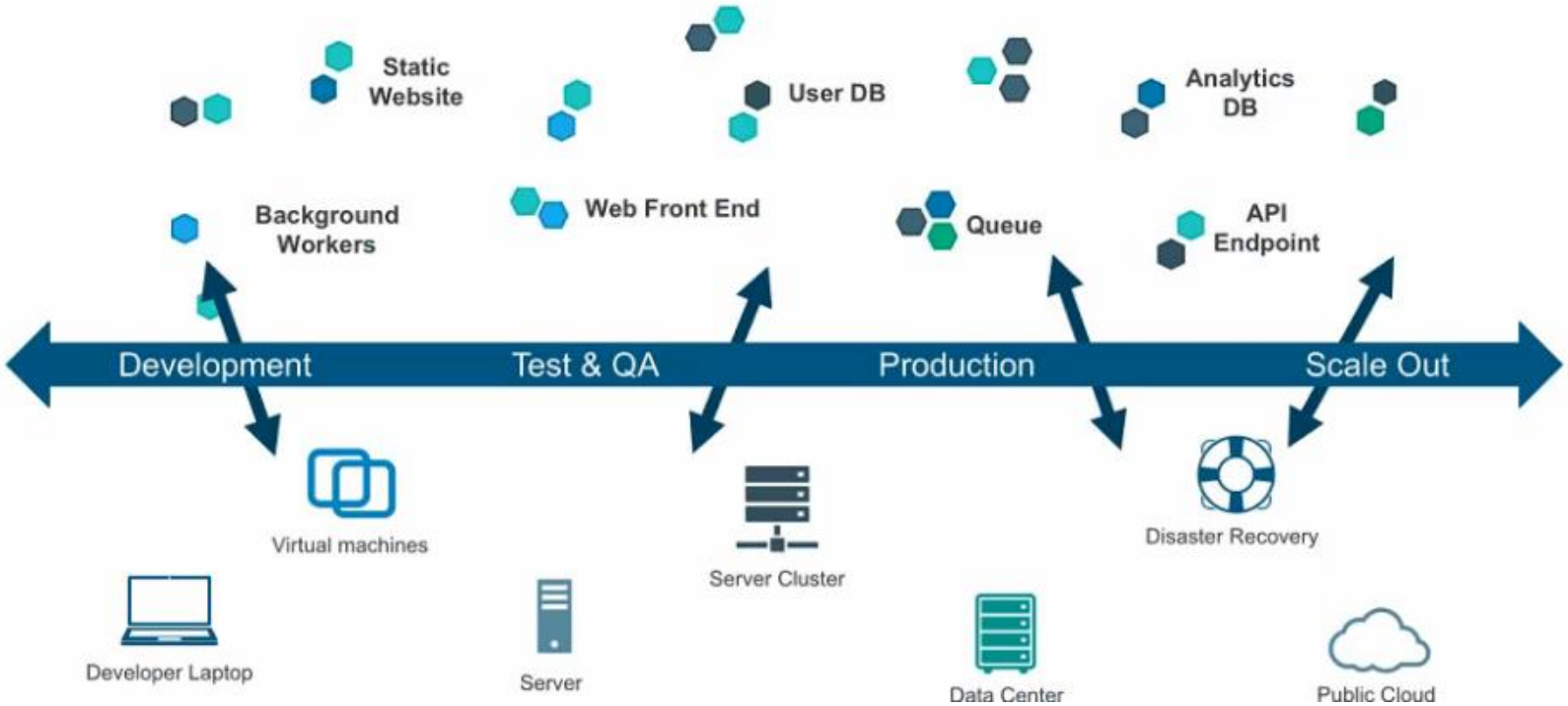
docker

Now....much more distributed, complex..

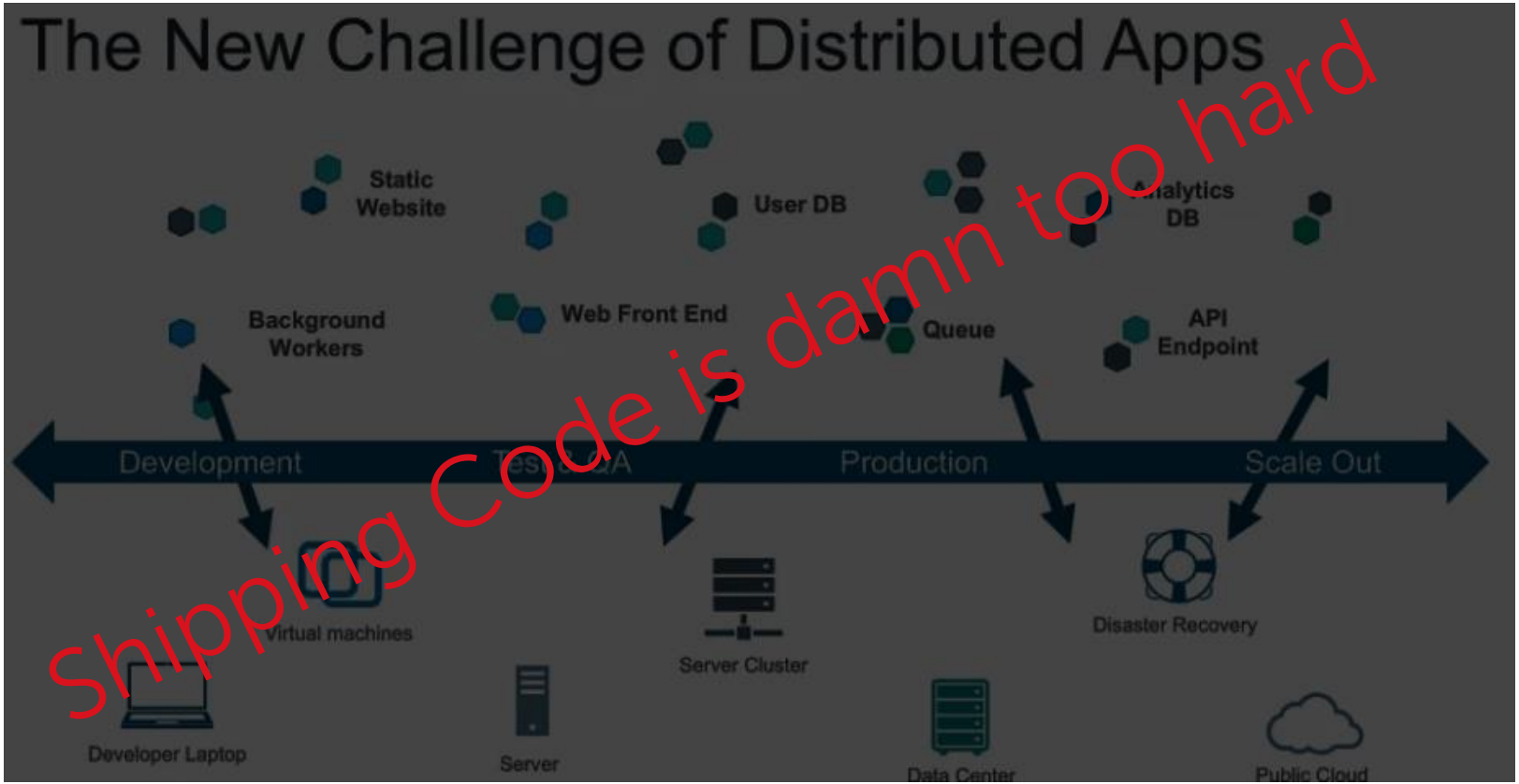


docker

The New Challenge of Distributed Apps



docker

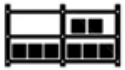

The New Challenge of Distributed Apps
Static Website
User DB
Analytics DB
Background Workers
Web Front End
Queue
API Endpoint
Development
Test & QA
Production
Scale Out
Virtual machines
Server Cluster
Disaster Recovery
Developer Laptop
Server
Data Center
Public Cloud
Shipping Code is damn too hard

docker

An Effort to solve the problem complexity…

docker

Every possible goods to ship X Every possible way to ship

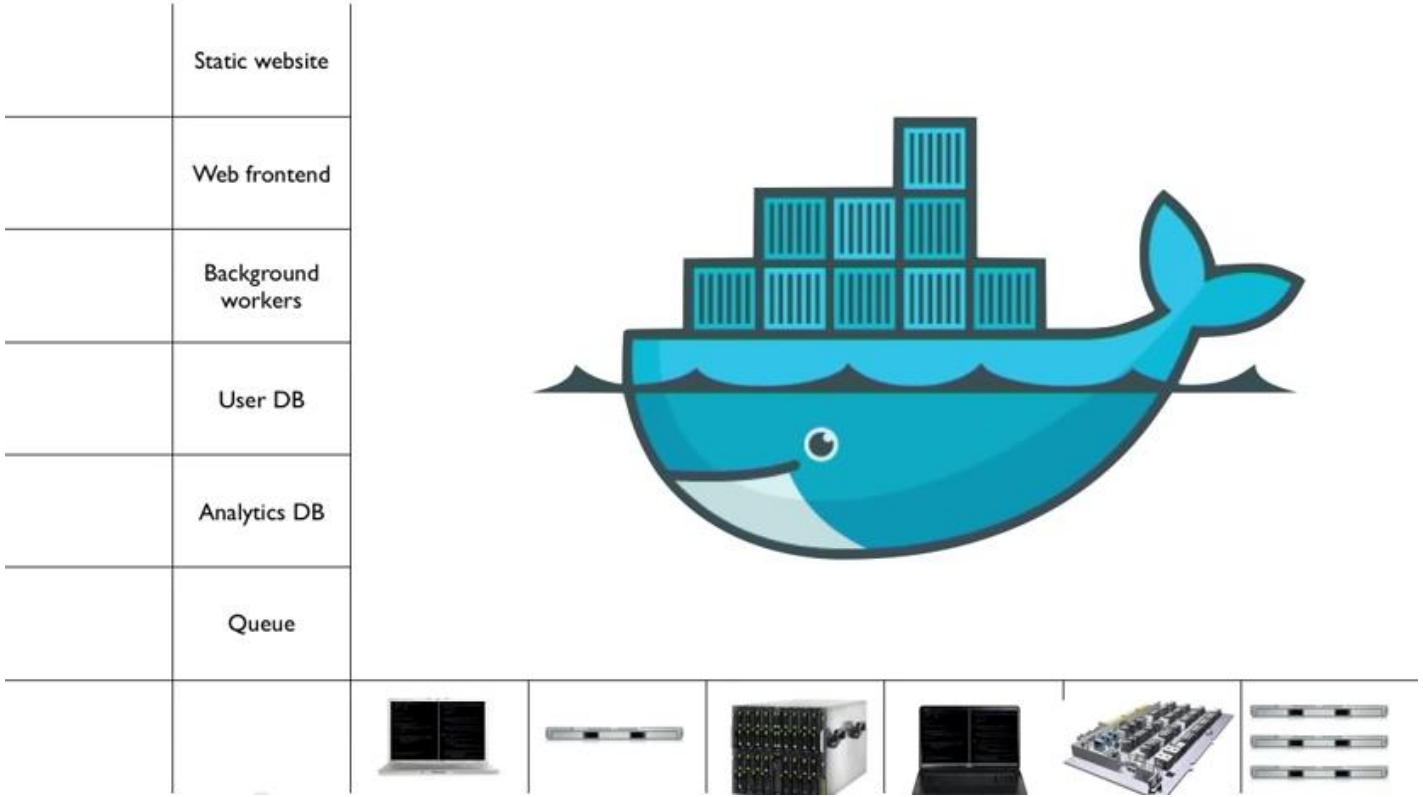
		?	?	?	?	?	?
		?	?	?	?	?	?
		?	?	?	?	?	?
		?	?	?	?	?	?
		?	?	?	?	?	?
		?	?	?	?	?	?
							

docker

A Solution…

docker

Docker ~ Brings standardization on packaging goods



docker

What is Docker?

Virtualization Tool?

VirtualBox

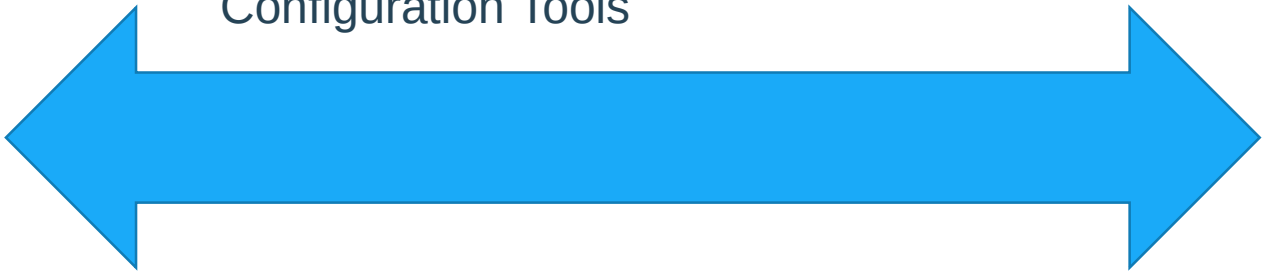
Configuration Manager?

VM Manager?

Less Portable,
Minimal Overhead

Most Portable,
Lots of Overhead

Configuration Tools



Manual Configuration

Traditional VMs

Less Portable,
Minimal Overhead

Most Portable,
Lots of Overhead

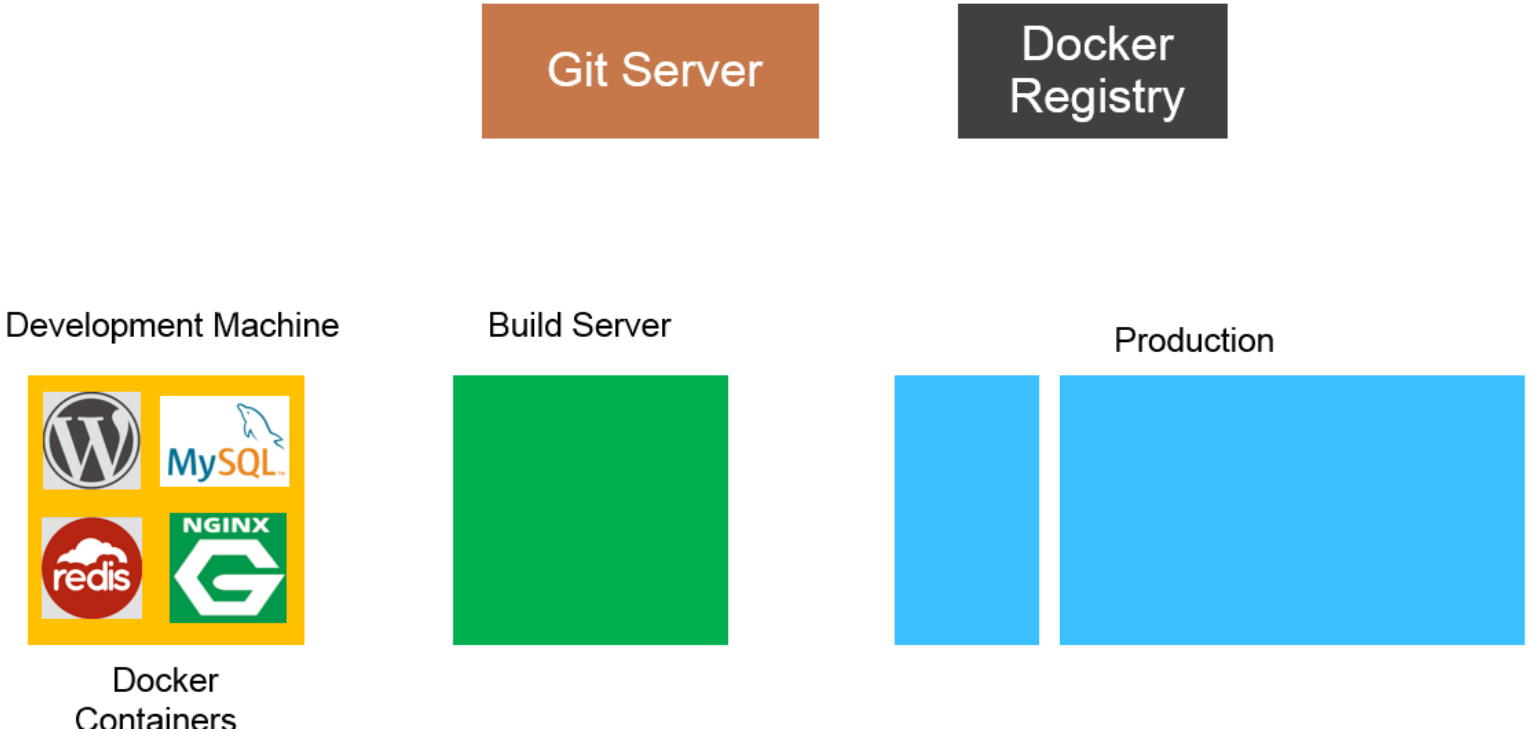
Configuration Tools

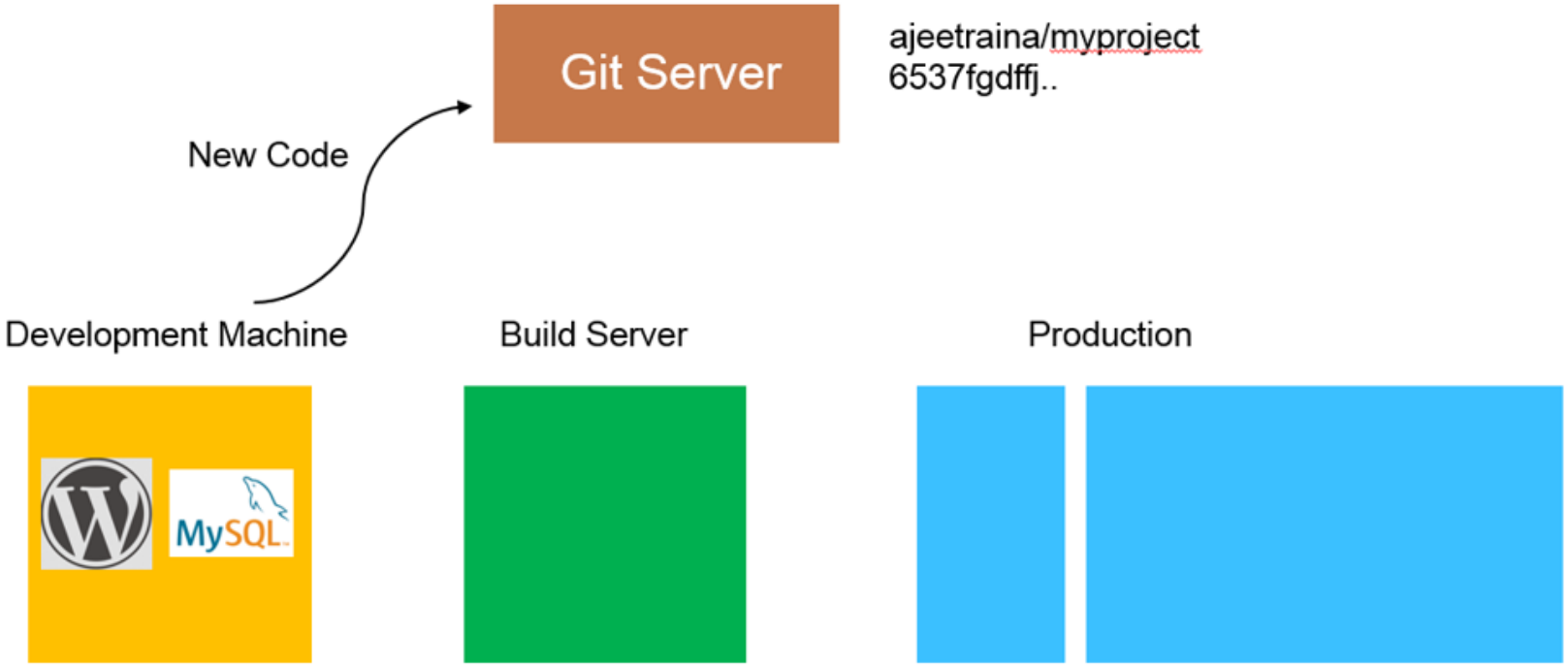
docker

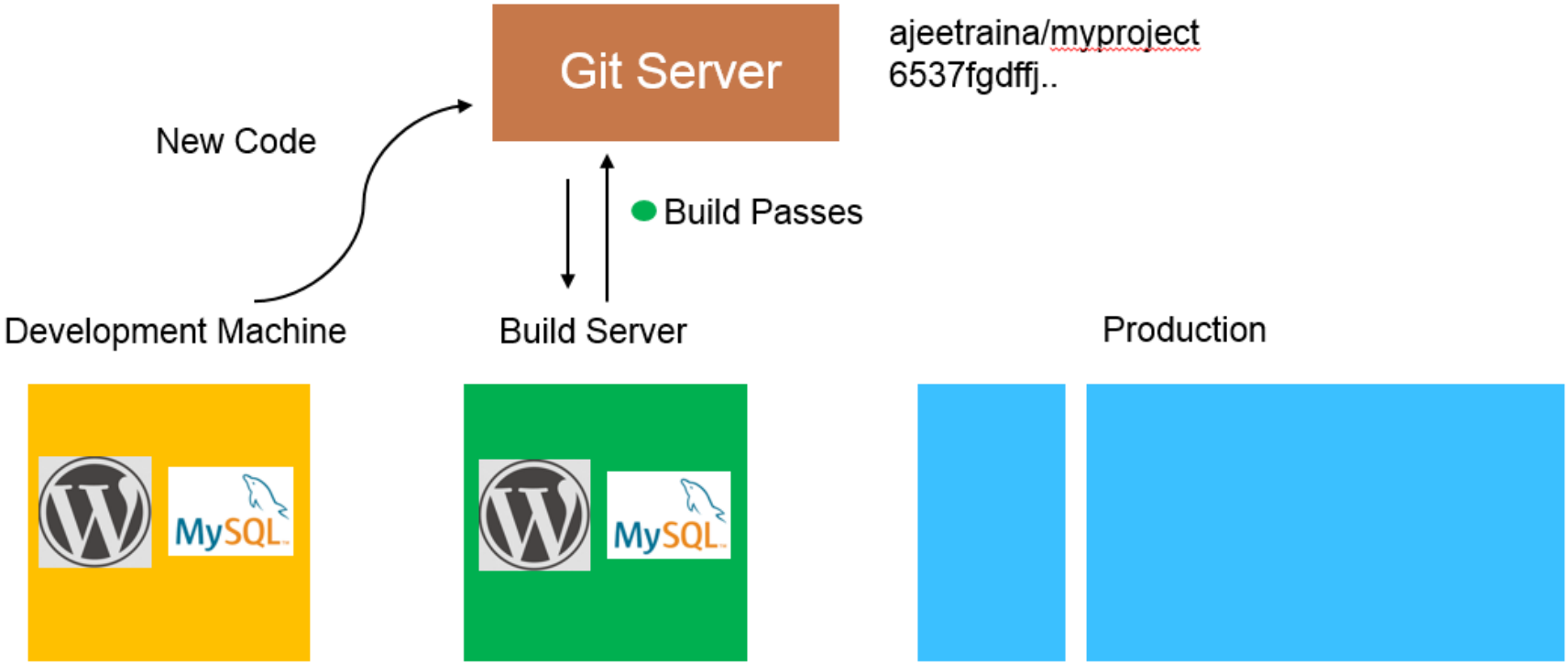
Manual Configuration

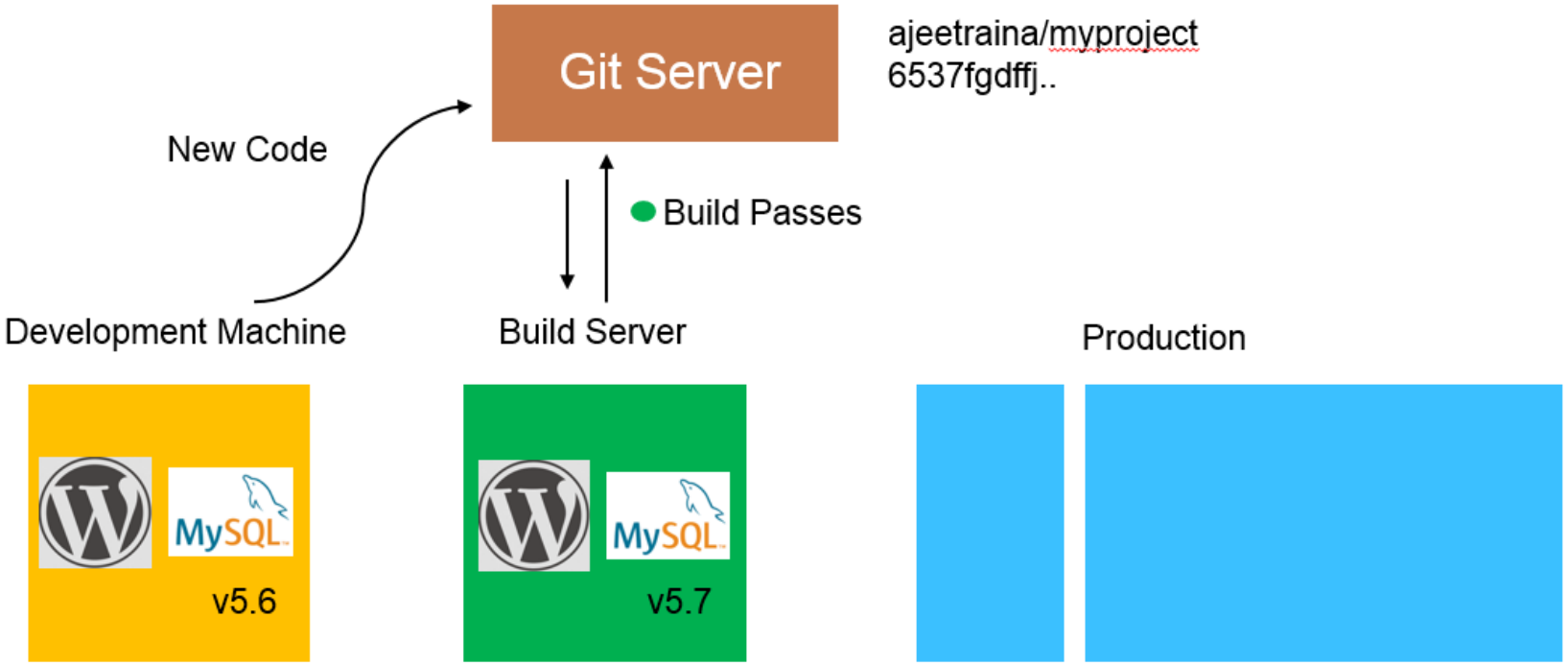
Traditional VMs

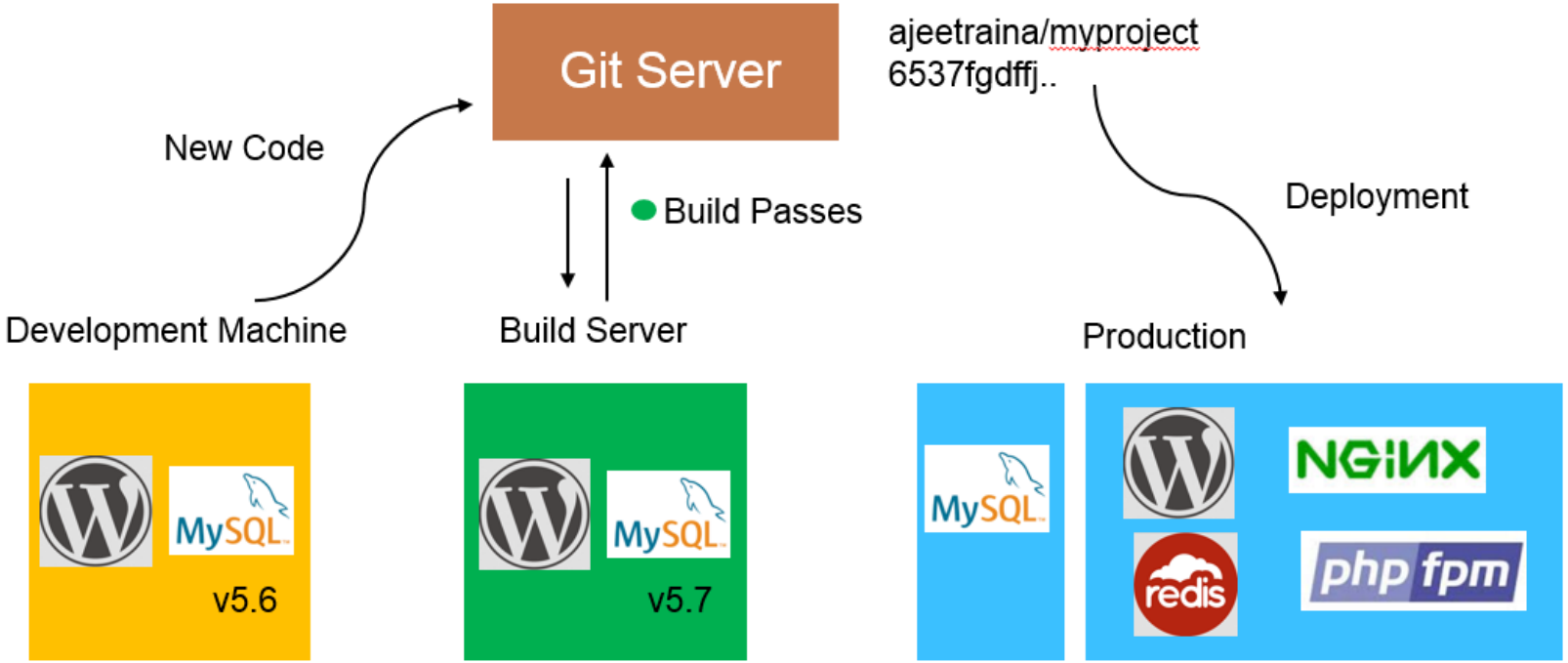
Traditional Software Development Workflow (Without Docker)


Git Server
Docker Registry
Development Machine
Build Server
Production
MySQL
NGINX
redis
Docker Containers

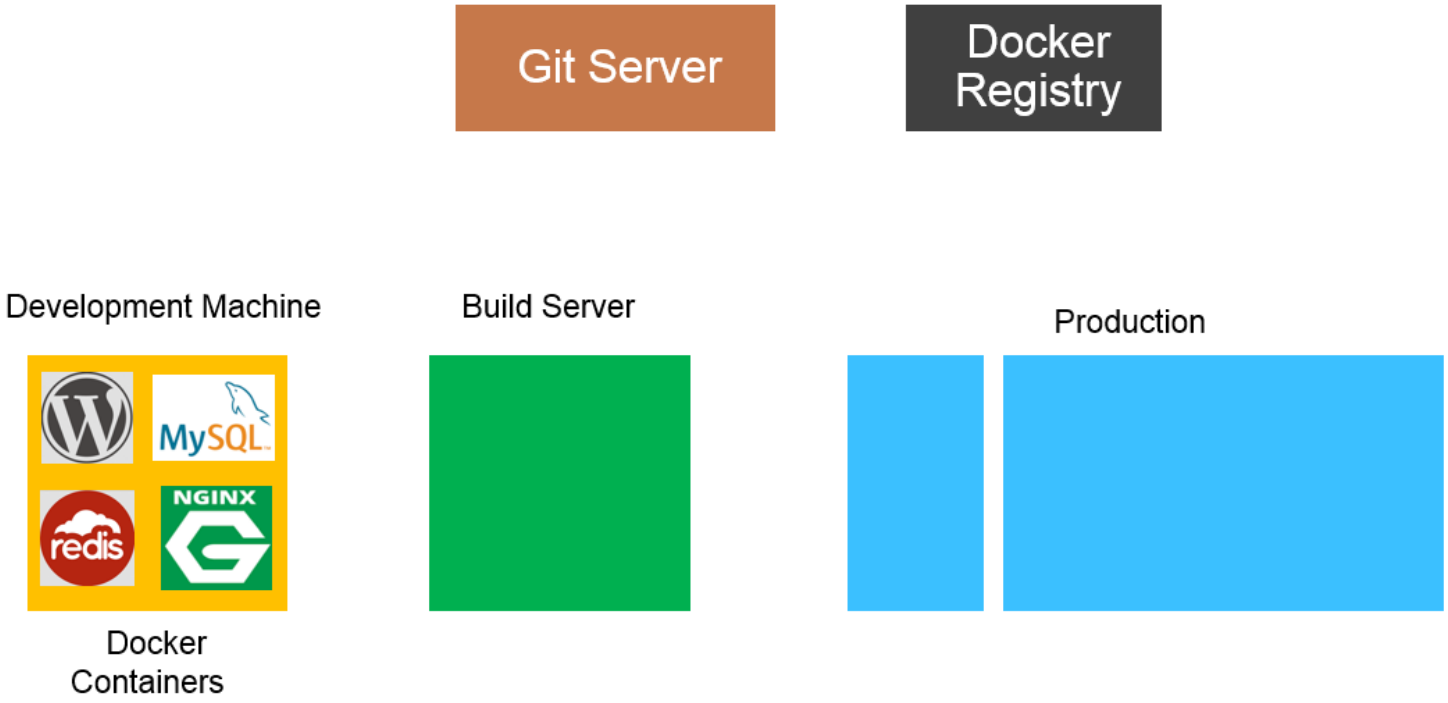

Git Server
ajeetraina/myproject
6537fgdffj..
New Code
Development Machine
Build Server
Production
MySQL

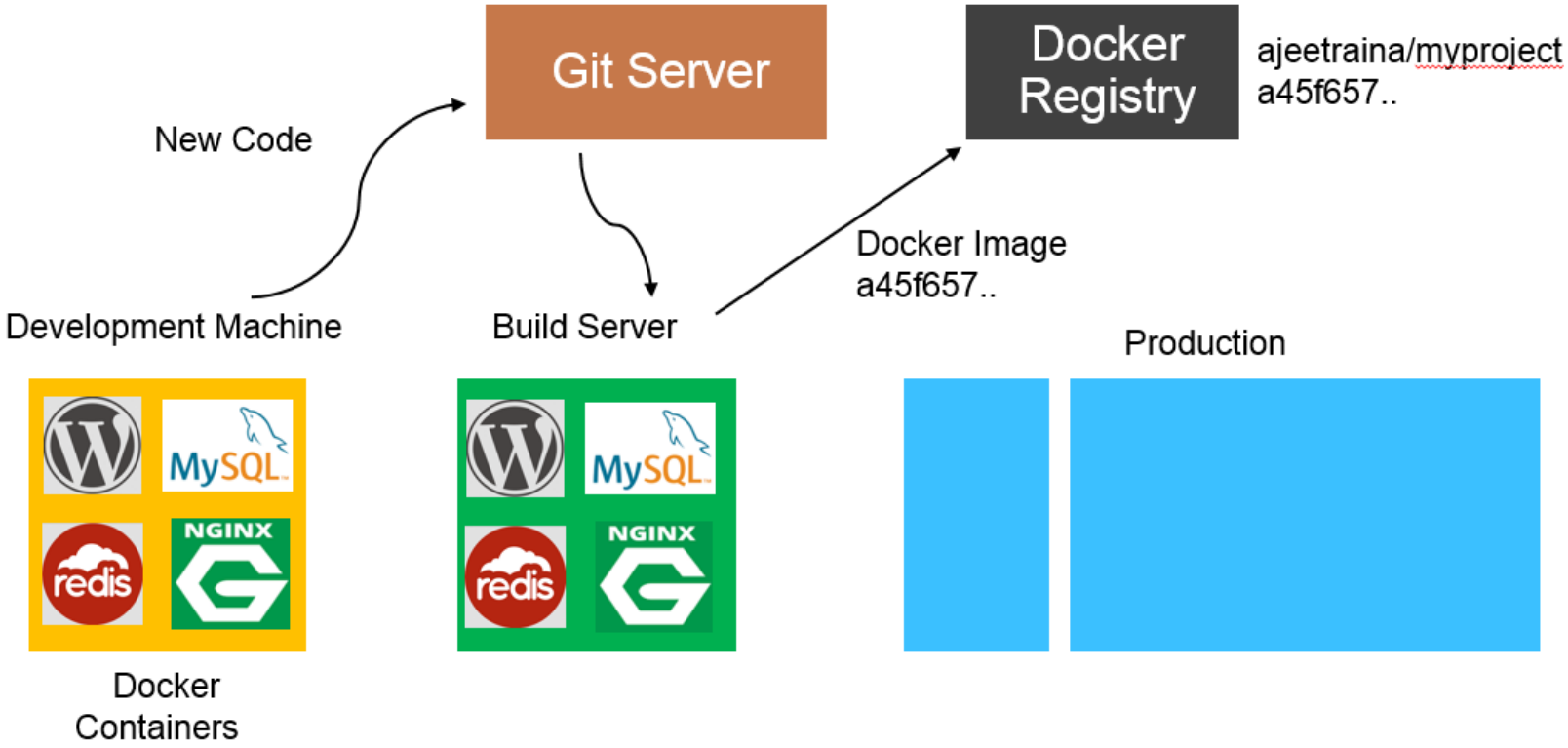

Git Server
ajeetraina/myproject
6537fgdffj..
New Code
Build Passes
Development Machine
Build Server
Production
MySQL
MySQL

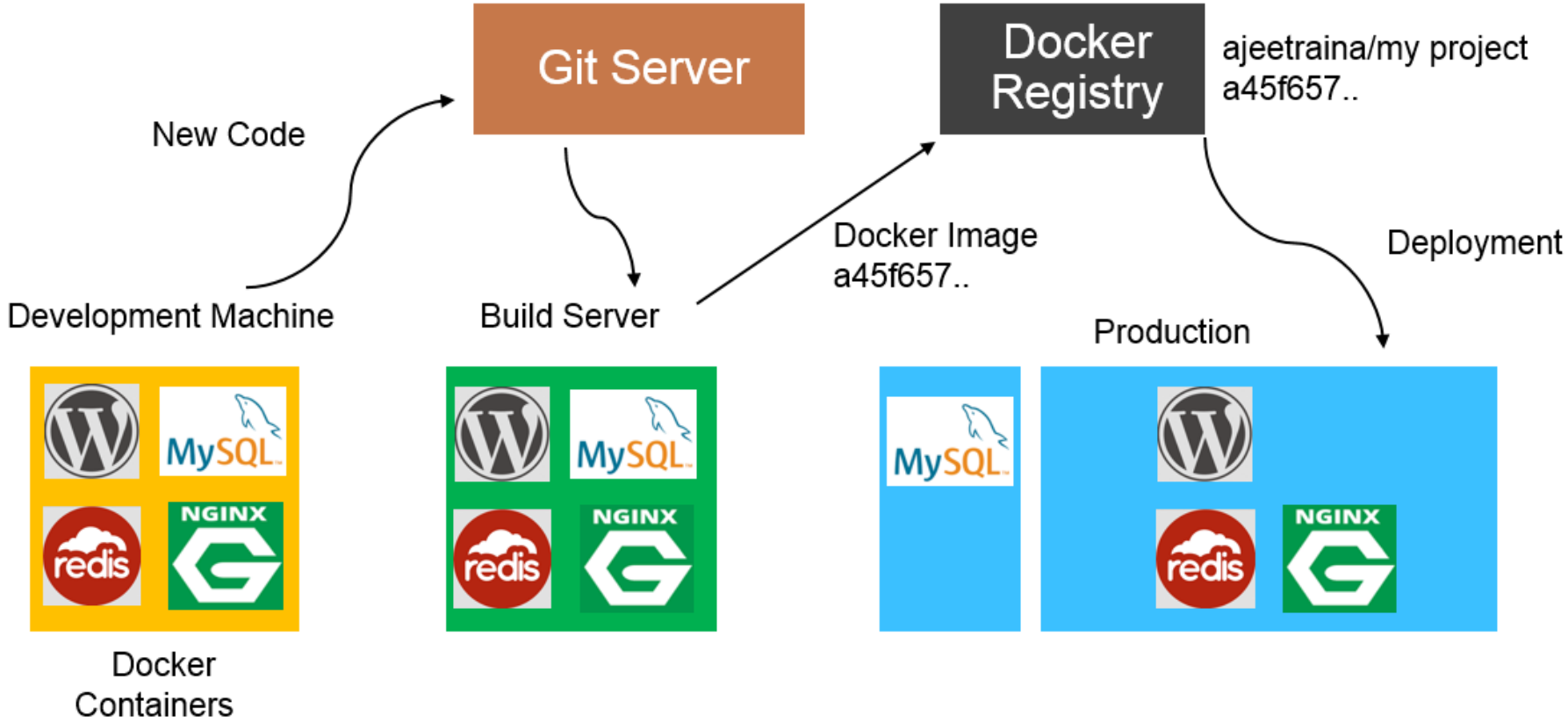

Git Server
ajeetraina/myproject
6537fgdffj..
New Code
Build Passes
Development Machine
Build Server
Production
MySQL
v5.6
MySQL
v5.7


Git Server
ajeetraina/myproject
6537fgdffj..
New Code
Build Passes
Deployment
Development Machine
Build Server
Production
MySQL
v5.6
MySQL
v5.7
MySQL
NGINX
redis
php fpm

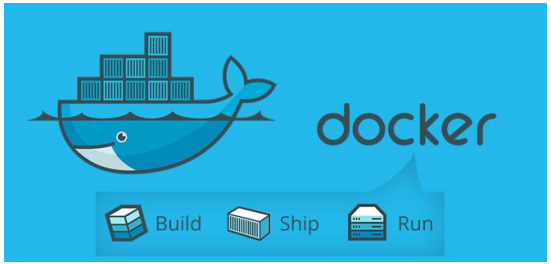
Traditional Software Development Workflow (With Docker)


Git Server
Docker Registry
Development Machine
Build Server
Production
MySQL
NGINX
redis
Docker Containers


Git Server
Docker Registry
ajeetraina/myproject
a45f657..
New Code
Docker Image
a45f657..
Development Machine
Build Server
Production
MySQL
NGINX
redis
Docker Containers


Git Server
Docker Registry
ajeetraina/my project
a45f657..
New Code
Docker Image
a45f657..
Deployment
Development Machine
Build Server
Production
MySQL
NGINX
redis
Docker
Containers

What is Docker?

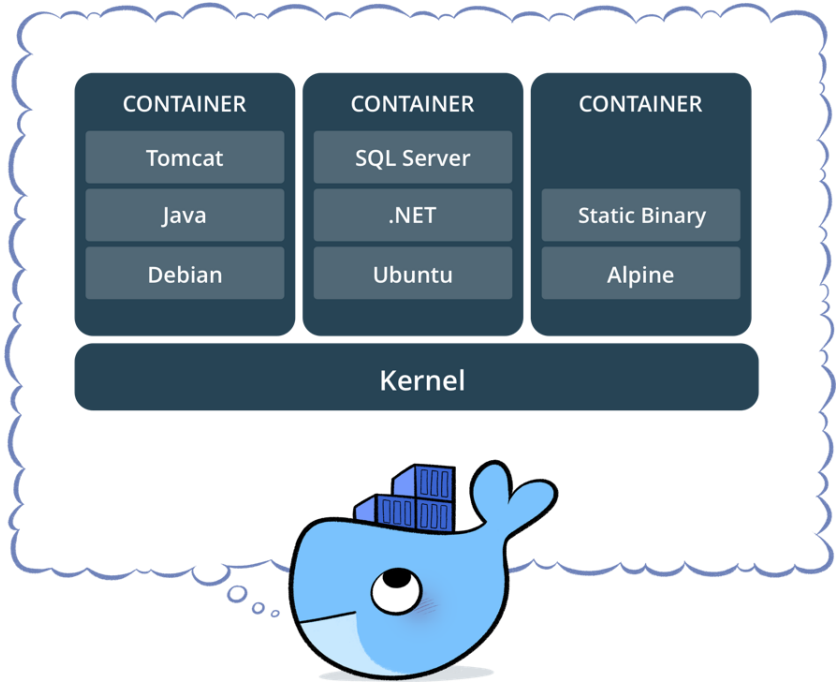


- A Company as well as Product
- Developed by DotCloud Inc. (Currently Docker Inc.)
- A Framework they built their PaaS upon.
- Released it as open source 4+ years back
- Cross-platform nature and friendliness towards System Admin and Developers
- Developers - concentrate on building applications and getting them running inside the containers
- System Admins - work on running the containers in deployment.
- Possible to set up in any OS, be it Windows, OSX, Linux - It work the same way
- Guaranteed to run the same way - Your development desktop, a bare-metal server, virtual machine, data center, or cloud

What is Docker?

- A tool that can package an application and its dependencies in a virtual container.
- Implementation of a container which is portable using a concept of image
- Docker uses host OS kernel, there is no custom or additional kernel inside container. All containers runs on machine are sharing this "host" kernel.
- Docker uses resource isolation features of the Linux kernel such as cgroups and kernel namespaces to allow independent "containers" to run within a single Linux instance, avoiding the overhead of starting virtual machines.

What is Docker?



- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works for all major Linux distributions
- Containers native to Windows Server 2016

docker

Docker containers are NOT VMs

Docker containers are NOT VMs

- It’s not quite like a VM
- Uses the host kernel
- Can’t boot a different OS
- Can’t have its own modules
- Doesn’t need init as PID 1
- Doesn’t need syslogd, cron.

It’s just a normal process on the host machine

- Contrast with VMs which are opaque

VM Vs Docker - Similarity

Virtual Machines	Docker
Process in one VM can’t see processes in other VMs	Process in one container can’t see processes in other container
Each VM has its own root filesystem	Each container has its own root file system(Not Kernel)
Each VM gets its own virtual network adapter	Docker can get virtual network adapter. It can have separate IP and ports
VM is a running instance of physical files(.VMX and .VMDK)	Docker containers are running instances of Docker Image
Host OS can be different from guest OS	Host OS can be different from Container OS

VM Vs Docker - Difference

Virtual Machines	Docker
Each VM runs its own OS	All containers share the same Kernel of the host
Boot up time is in minutes	Containers instantiate in seconds
VMs snapshots are used sparingly	Images are built incrementally on top of another like layers. Lots of images/snapshots
Not effective diffs. Not version controlled	Images can be diffed and can be version controlled. Dockerhub is like GITHUB
Cannot run more than couple of VMs on an average laptop	Can run many Docker containers in a laptop.
Only one VM can be started from one set of VMX and VMDK files	Multiple Docker containers can be started from one Docker image

docker

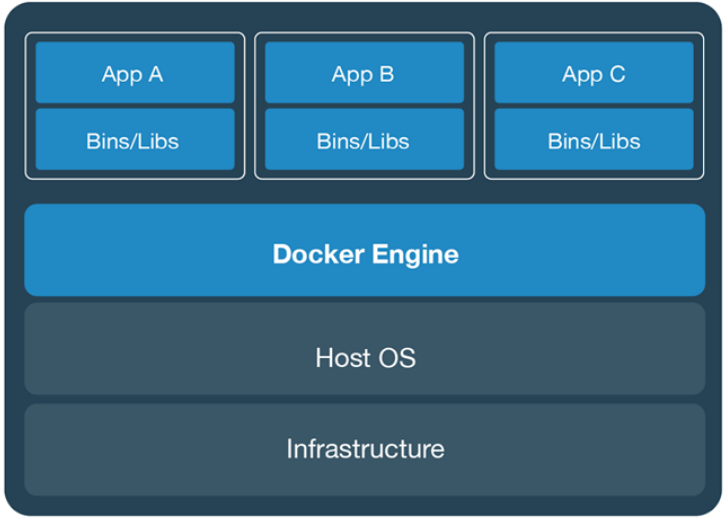
VMs

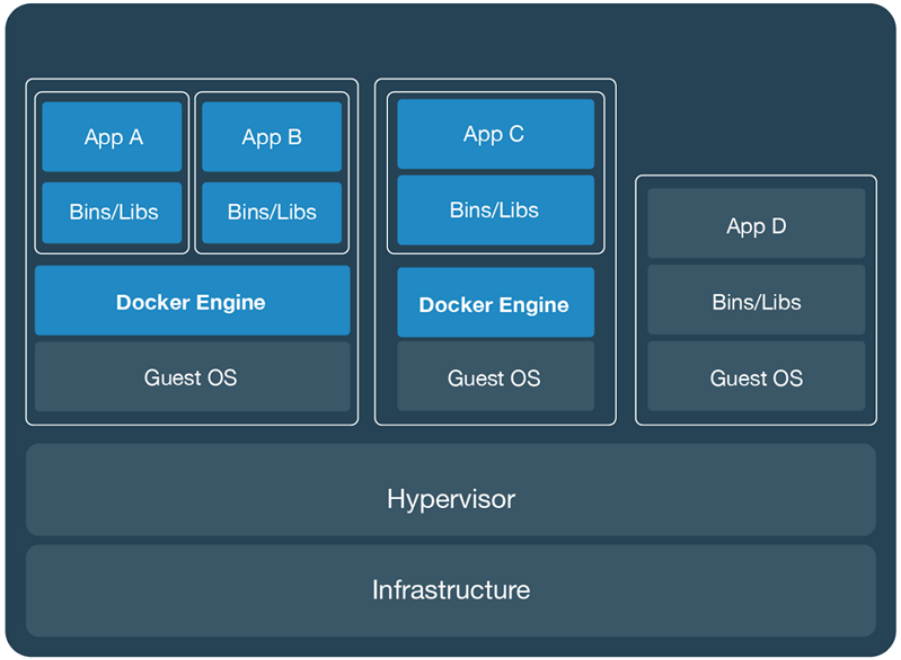
docker

Containers

docker

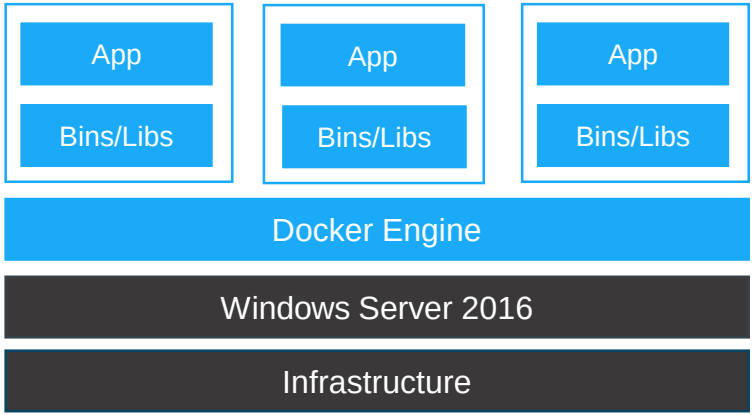
Containers Vs VMs





docker

Docker + Windows Server = Windows Containers



Native Windows containers powered by Docker Engine

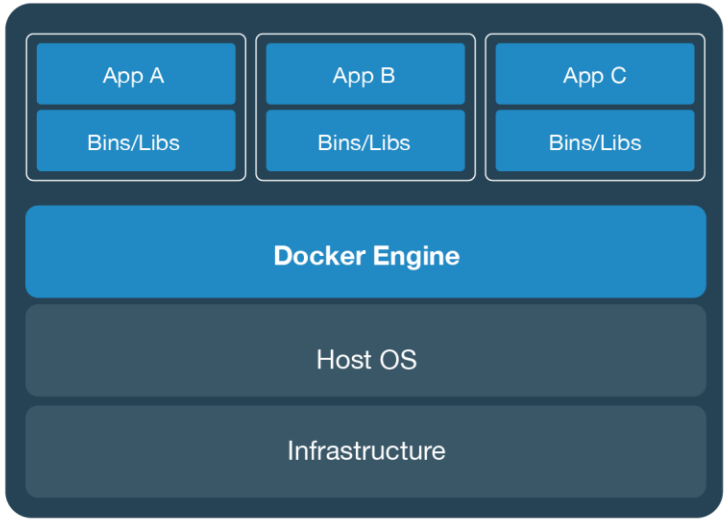
Windows kernel engineered with new primitives to support containers

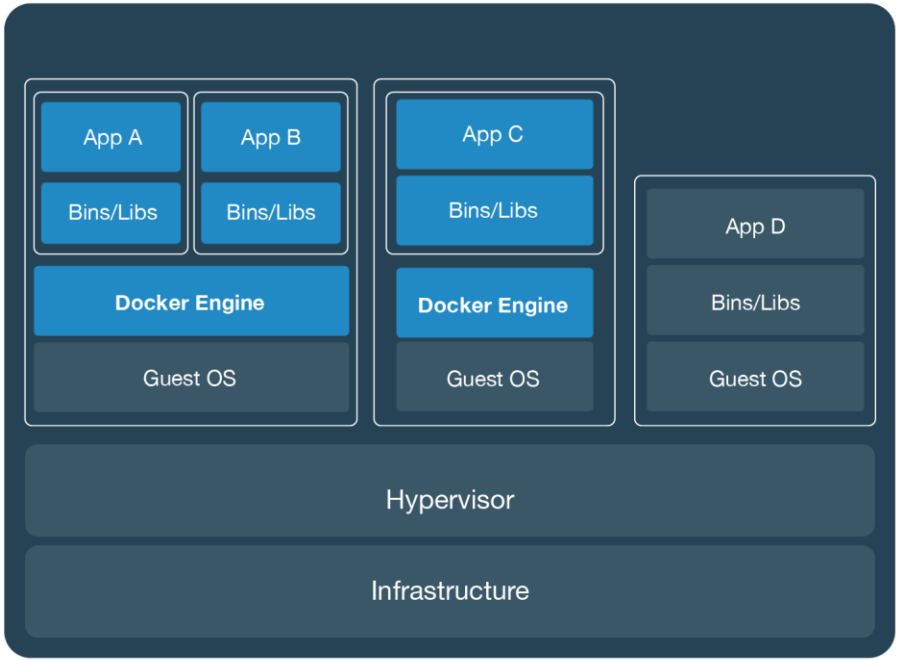
Deep integration with 2+ years of engineering collaboration in Docker Engine and Windows Server

Microsoft is top 5 Docker open source project contributor and a Docker maintainer

docker

They're different, not mutually exclusive





docker

Docker Vocabulary

Some Docker vocabulary

Containers	Images
How you **run** your application	How you **store** your application

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

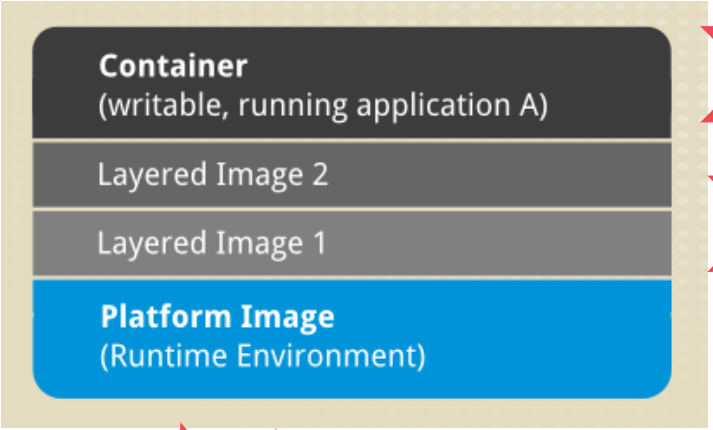
Docker Engine

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

Image Layering



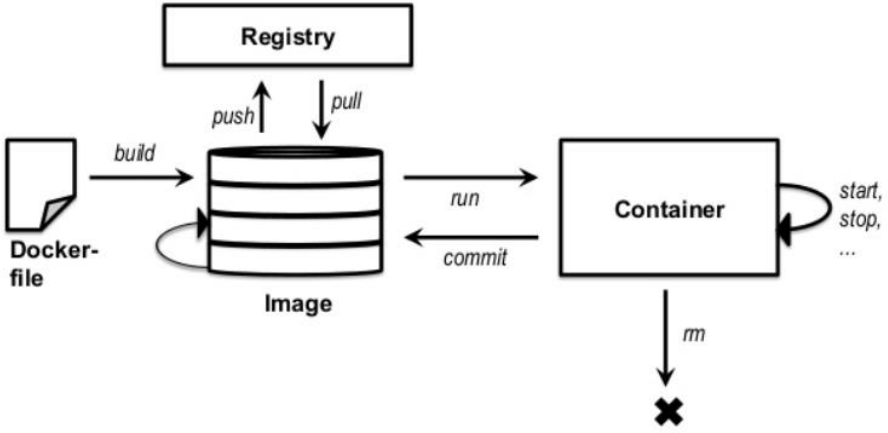
- An application sandbox.
- Each container is based on an image that holds necessary config data.
- When you launch a container from an image, a writable layer is added on top of this image

- A static snapshot of the containers' configuration.
- Image is a read-only layer that is never modified, all changes are made in top-most writable layer, and can be saved only by creating a new image.
- Each image depends on one or more parent images

- An image that has no parent.
- Platform images define the runtime environment, packages and utilities necessary for containerized application to run.

docker

Basic Docker Commands



Pulling Docker Image

$ docker pull ajeetraina/catweb

Listing out Docker Images

$ docker image ls

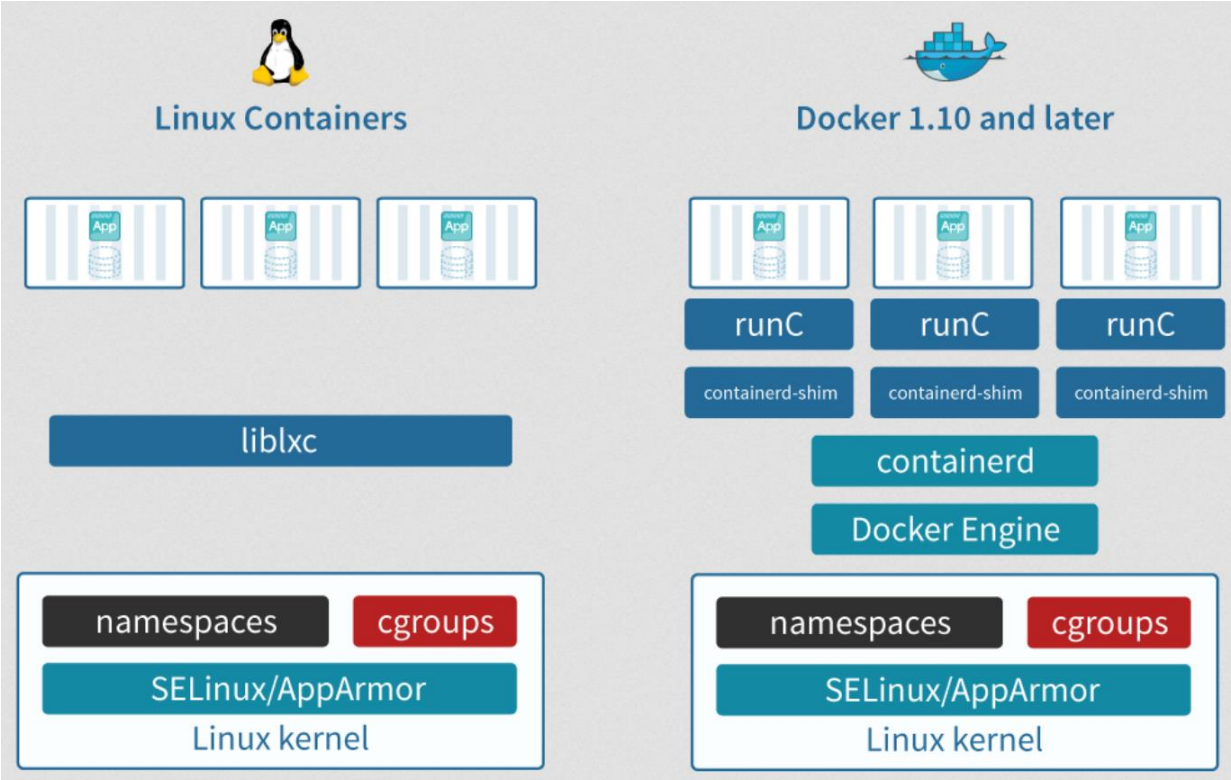
Running Docker Containers

$ docker container run –d –p 5000:5000 –-name catweb ajeetraina/catweb

Stopping the container

$ docker container stop catweb (or <container id>)

Docker Underlying Technology

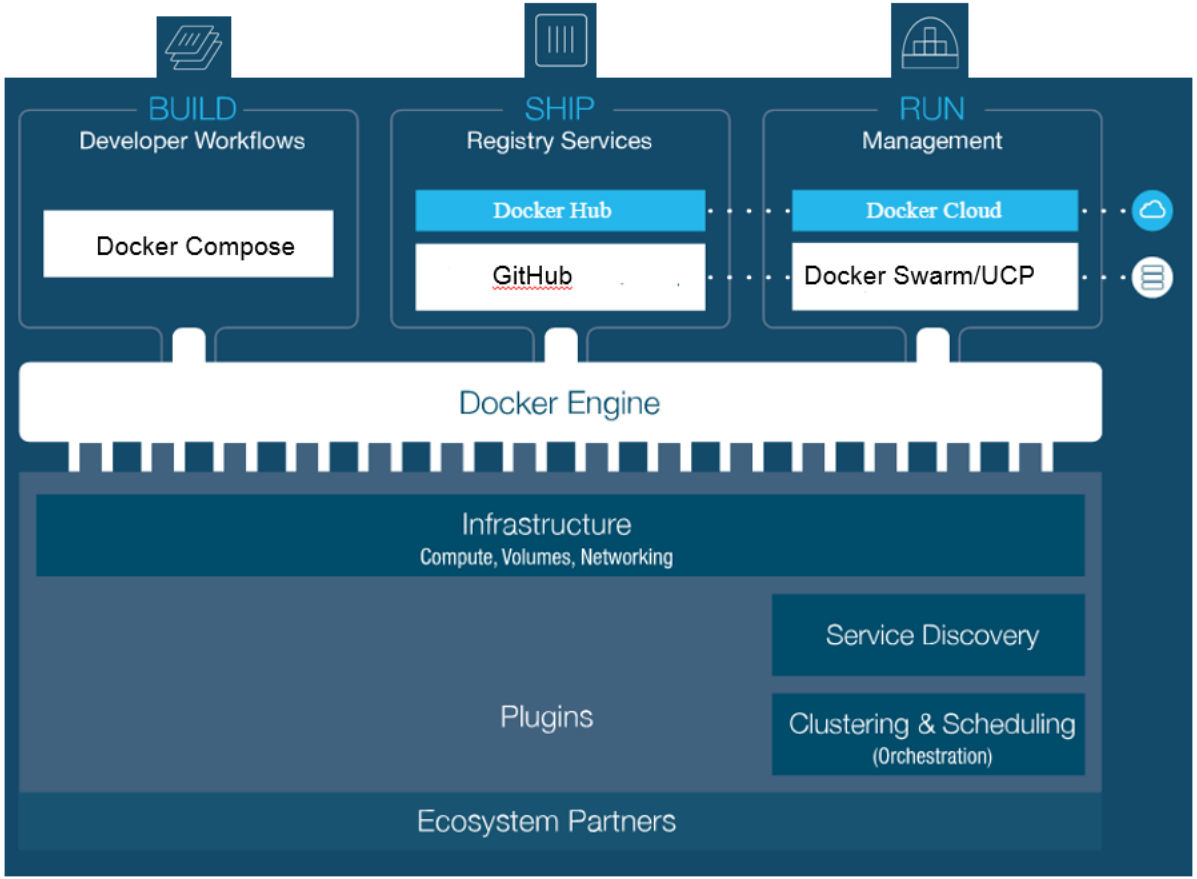


docker

Build, Ship & Run

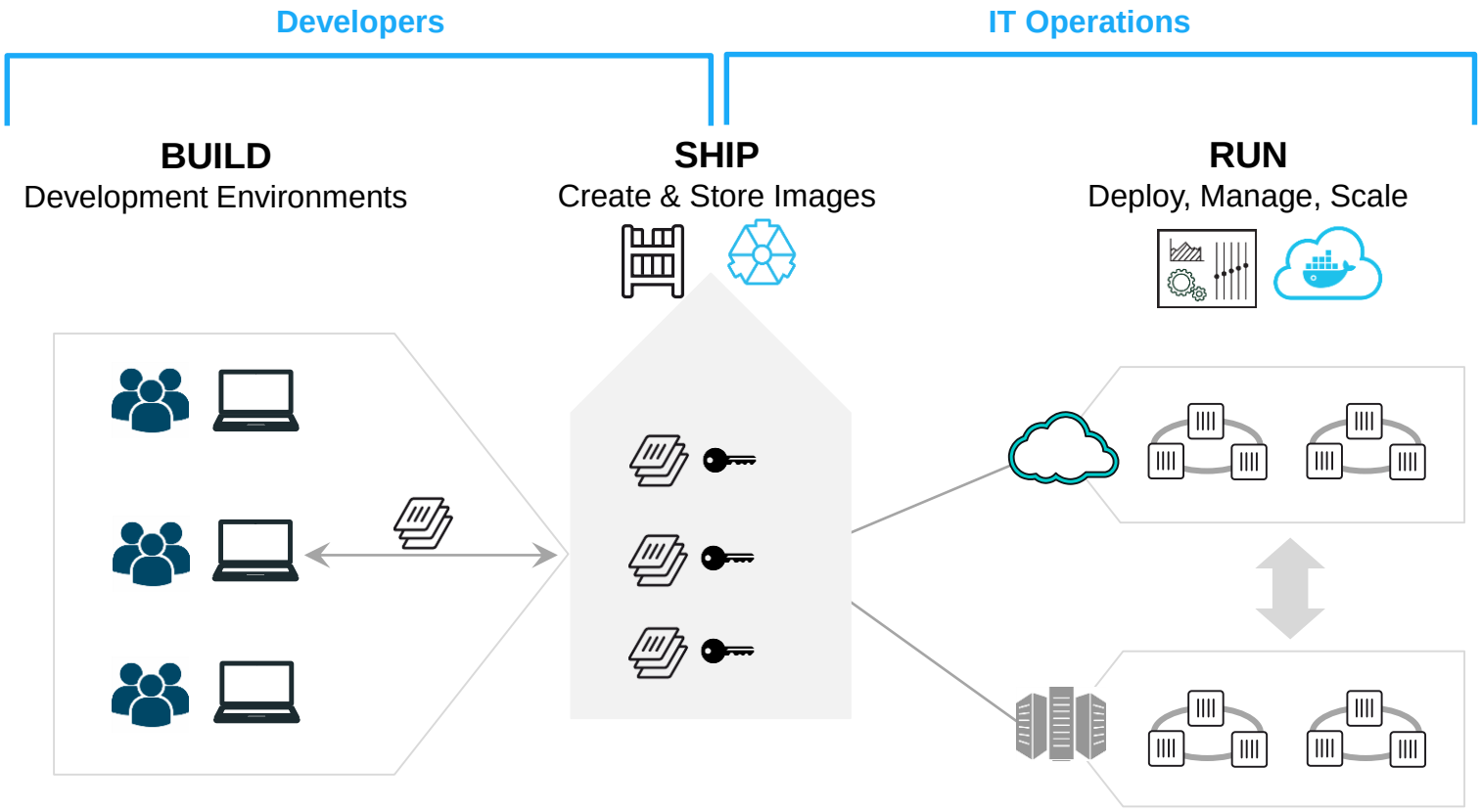
docker

Docker Mission



docker

Put it all together: Build, Ship, Run Workflow

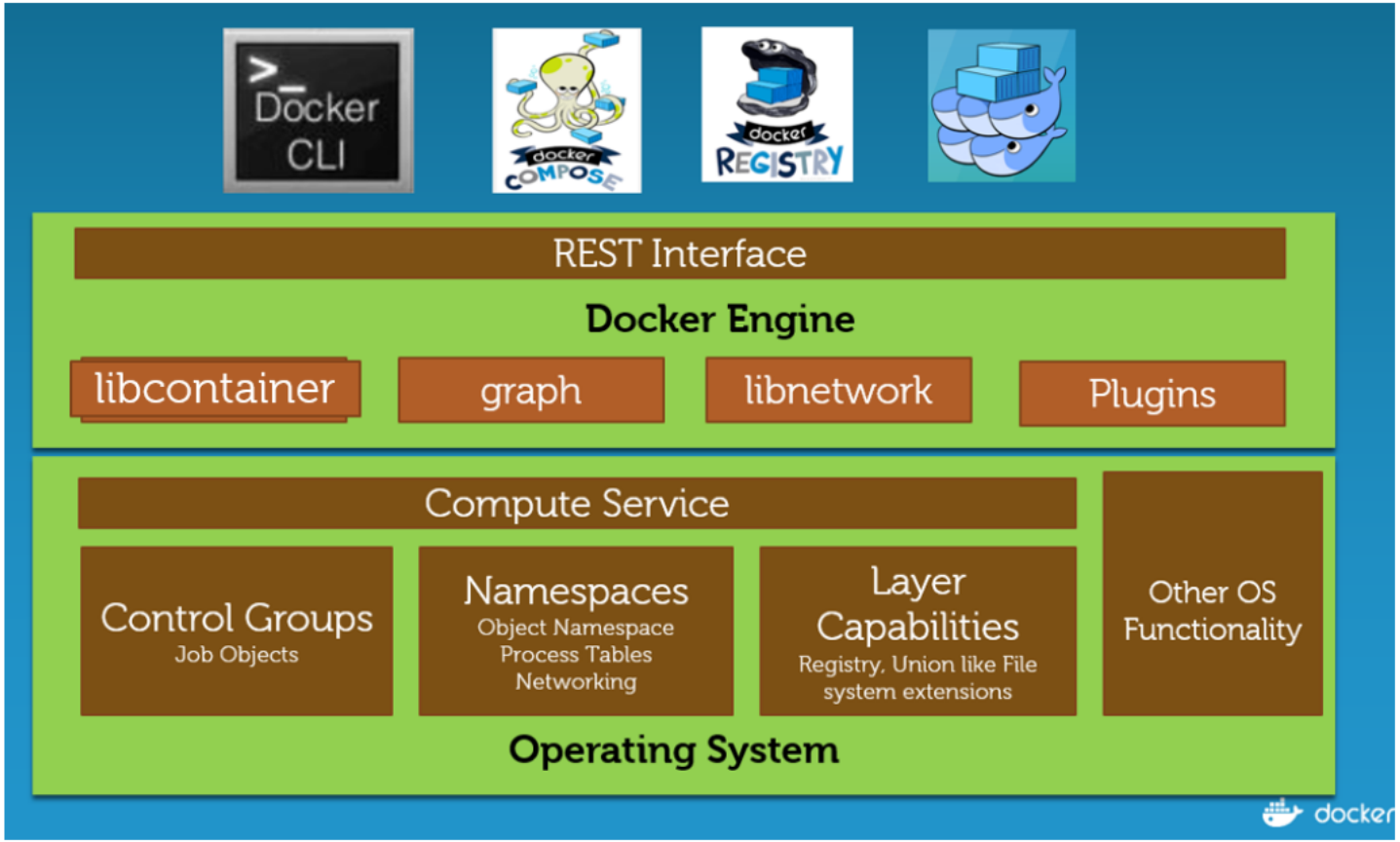


docker

Docker on Linux Vs Windows

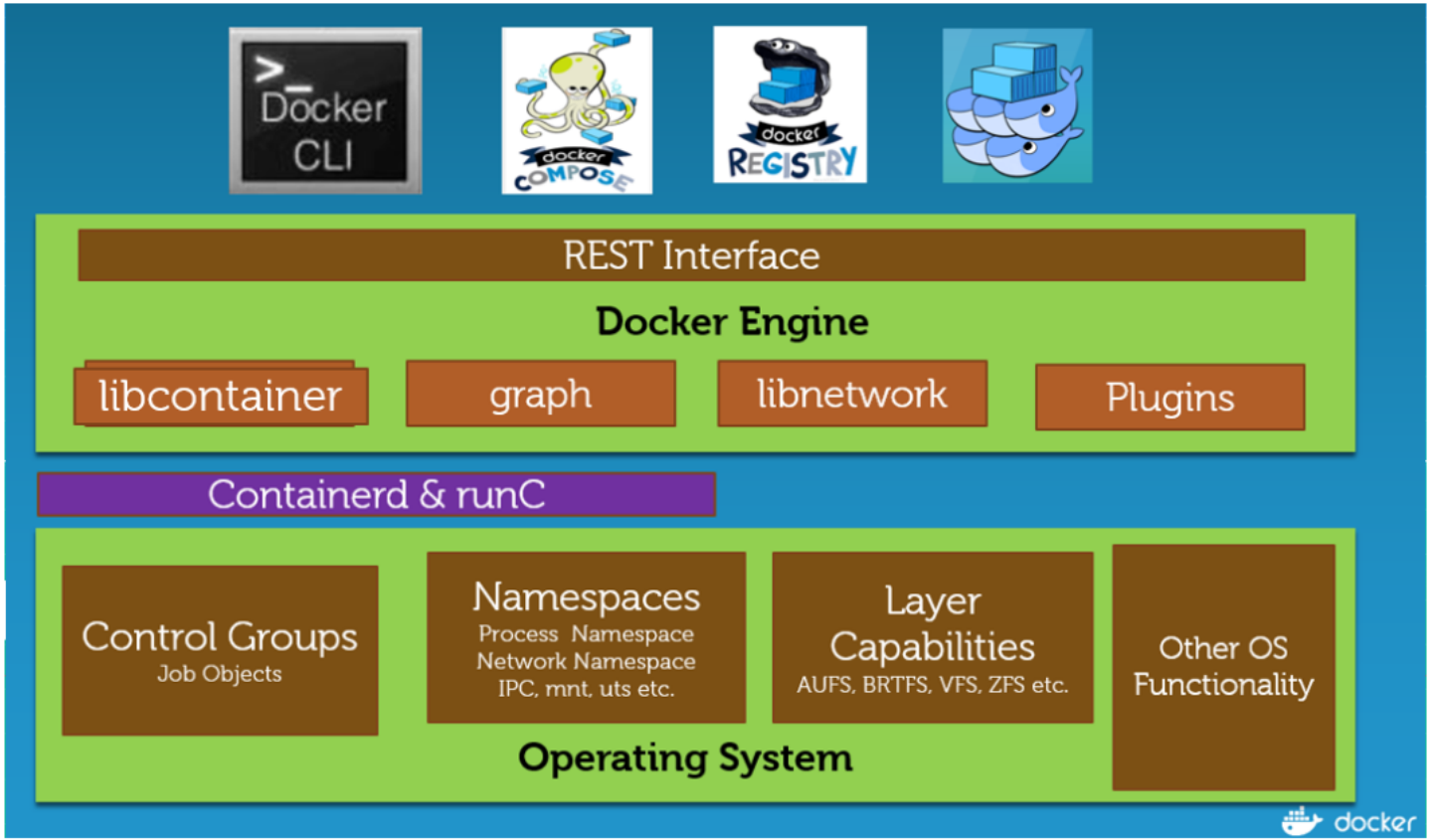
docker

Docker Engine on Windows Platform



docker

Docker Engine on Linux Platform



docker

Dockerfile – Windows Example

```
PS C:\Users\Ajeet_Raina\Desktop> cat .\Dockerfile
# This dockerfile utilizes components licensed by their respective owners/authors.

FROM microsoft/windowsservercore

LABEL Description="IIS" Vendor="Microsoft" Version="10"

RUN powershell -Command Add-WindowsFeature Web-Server

CMD [ "ping", "localhost", "-t" ]
```

Dockerfile – Linux Example

```
FROM ajeetraina/apache
MAINTAINER Ajeet Raina <ajeetraina@gmail.com>

RUN apt-get update && apt-get -y install php5 php5-mysql && apt-get clean && rm -rf /var/lib/apt/lists/*
RUN /usr/sbin/a2dismod 'mpm_*' && /usr/sbin/a2enmod mpm_prefork
EXPOSE 80
EXPOSE 443

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

docker

Docker Compose – Building Microservices in easy way

```yaml
version: '3'
services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress
   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     ports:
       - "8000:80"
     restart: always
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
volumes:
    db_data:
```

Backend Service

Specify Volumes/Network

Environmental variables

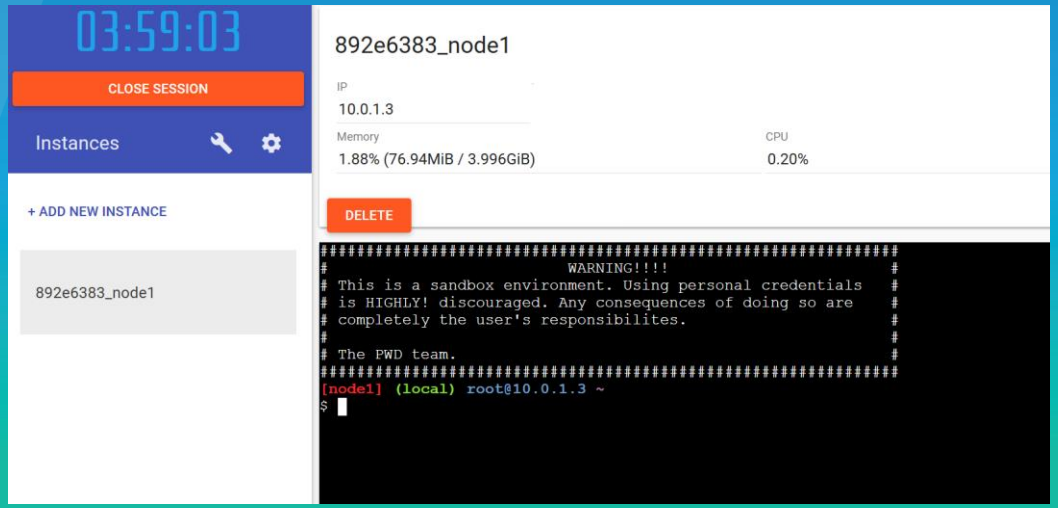
Frontend Service

Specify Volumes/Network

Environmental variables

docker

Demo



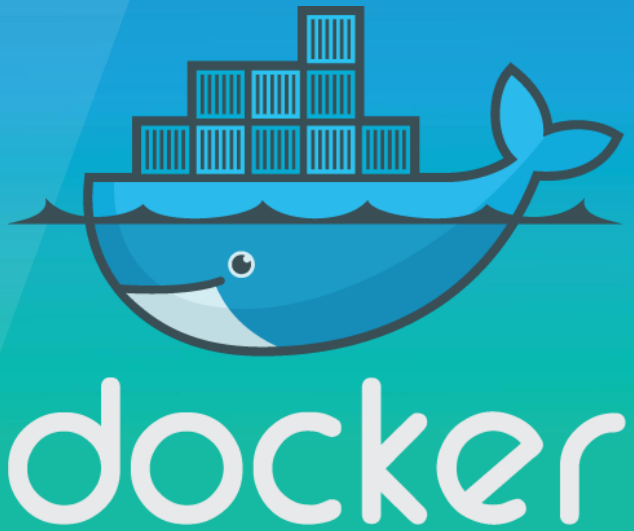
Docker Playground

References

www.collabnix.com
www.play-with-docker.com – Docker Playground (For Test Drive)
www.docs.docker.com

Thank You.

Questions?


docker