



docker

**DevOps Montréal**

Jan 6, 2014

**Colin Surprenant**

@colinsurprenant

[github.com/colinsurprenant/devopsmtl-docker](https://github.com/colinsurprenant/devopsmtl-docker)

# What is Docker?

Open source engine that leverage LXC and AUFS to package an application and its dependencies in a virtual container that can run on any Linux server.

# That was easy!

Thank you!  
Good evening!



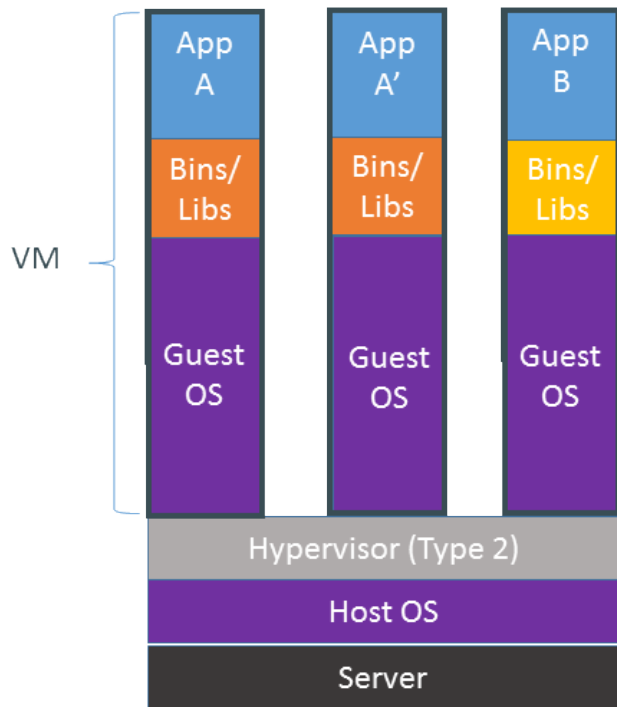
# What is LXC?

- Linux Containers
- Available since kernel 2.6.27
- Not a new concept
  - Solaris Zones
  - FreeBSD Jails
  - Linux VServer
  - OpenVZ

# LXC

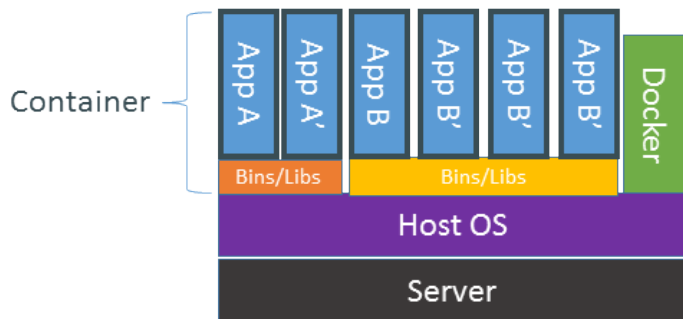
- Run Linux within Linux
- Lightweight VM
  - own process space
  - own network interface
  - SHARE kernel with host
- A container is a group of isolated processes
  - cgroups
  - namespace
- “chroot” on steroid

# LXC vs VM



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

...result is significantly faster deployment, much less overhead, easier migration, faster restart

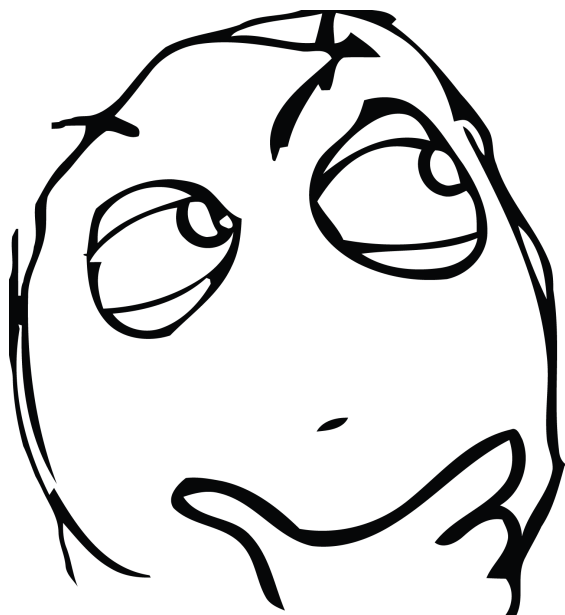


# LXC Performance?

Negligible overhead

- Isolated processes run straight on the host
- Native CPU performance
- Minimal memory overhead
- Minimal network performance overhead

# So then, what is Docker?



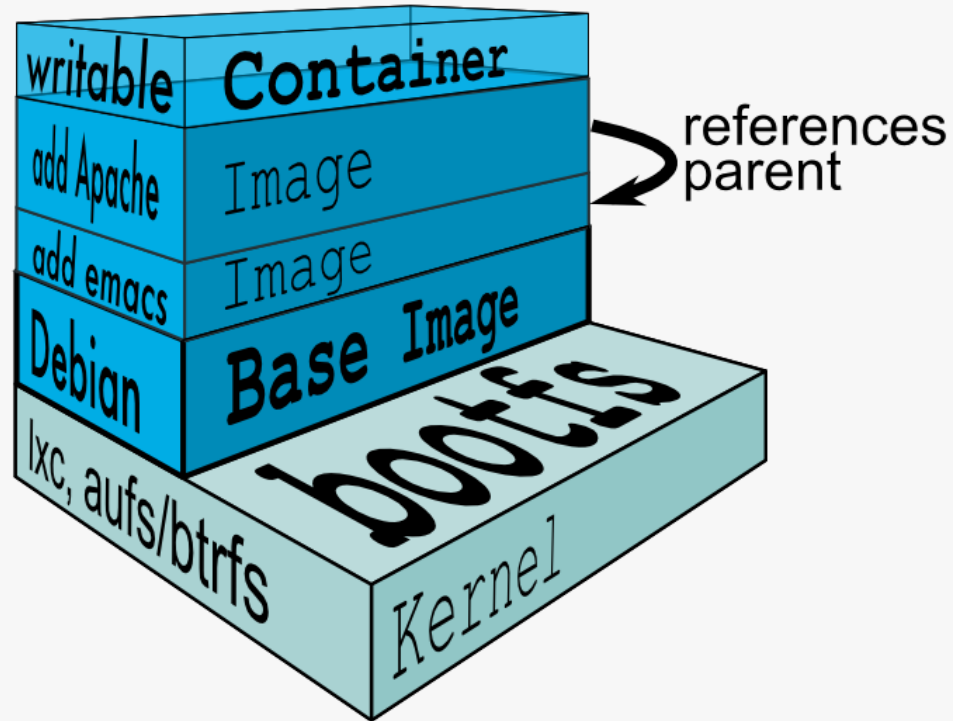


# Docker

- Tools to easily build images
- Share images using repositories
  - public repository: [index.docker.io](https://index.docker.io)
  - create your own private repository
- Docker daemon
  - manage containers & images
  - HTTP API
  - CLI tools

# Containers, images & AUFS

1. RO images depends on parent images
2. Add AUFS RW layer
3. All layers + meta is a container



# Why Docker?

- Containment and reproducibility
- Encapsulate app with its dependencies
- Run everywhere<sup>(TM)</sup>
- Another step toward *Immutable Infrastructure*

# Why Docker?

- No more missing dependencies in deployments
- Run side-by-side containers with own versions of dependencies

# Separation of Concerns

## **Developer**

**Inside** the container

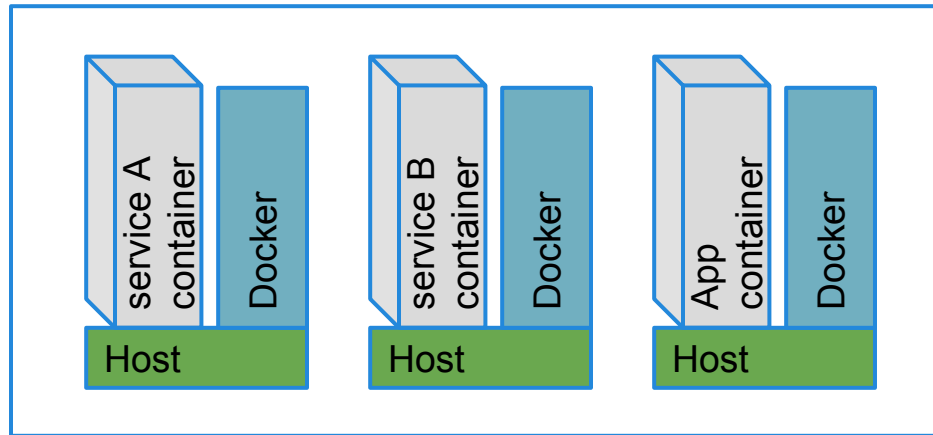
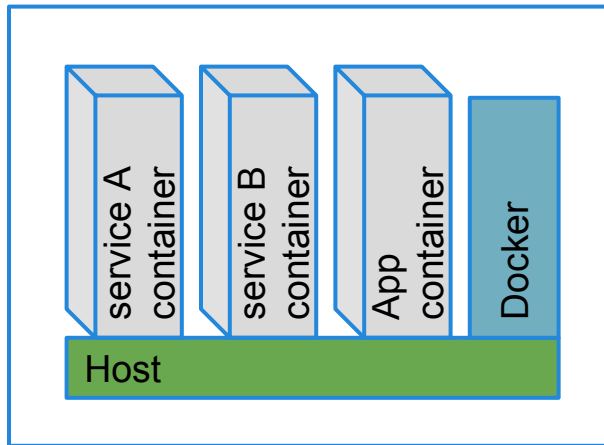
- my code
- my libraries
- my package manager
- my app
- my data

## **DevOps**

**Outside** the container

- logging
- remote access
- network configuration
- monitoring

# Architecture



# Some Consequences

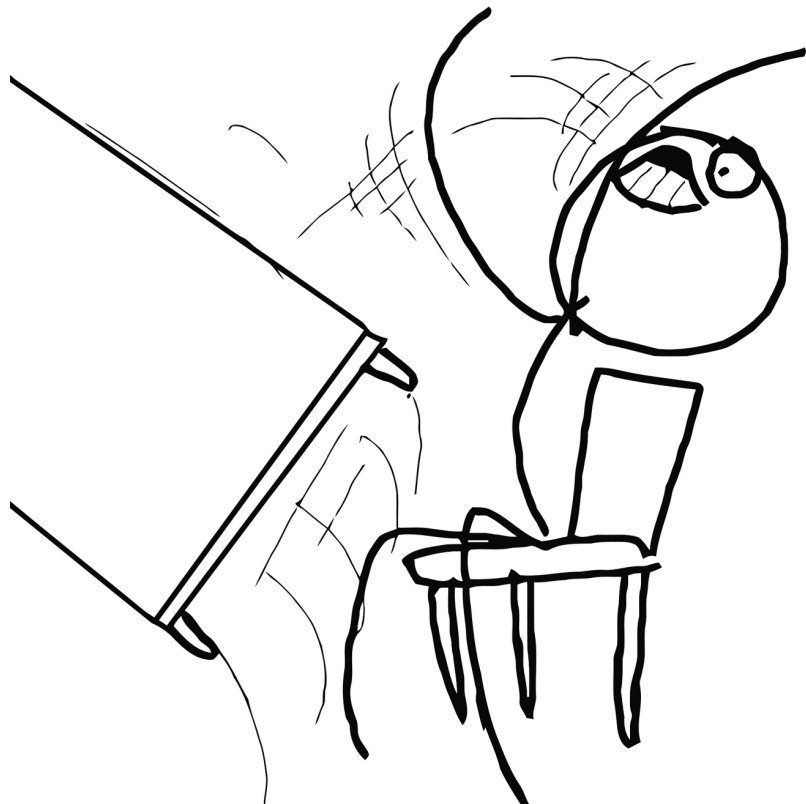
- Services discovery & interconnection logic
- “Dockerized” services management/lifecycle
- Services data persistence and logging
- Path to *The Twelve-Factors App*

# Requirements

- Linux kernel  $\geq 3.8$
- AUFS
- LXC
- 64 bits
- Recommended
  - Ubuntu 12.04 with upgraded kernel
  - Ubuntu 13.04



# Bro, I'm developing on a MacBook



WTF MacBook

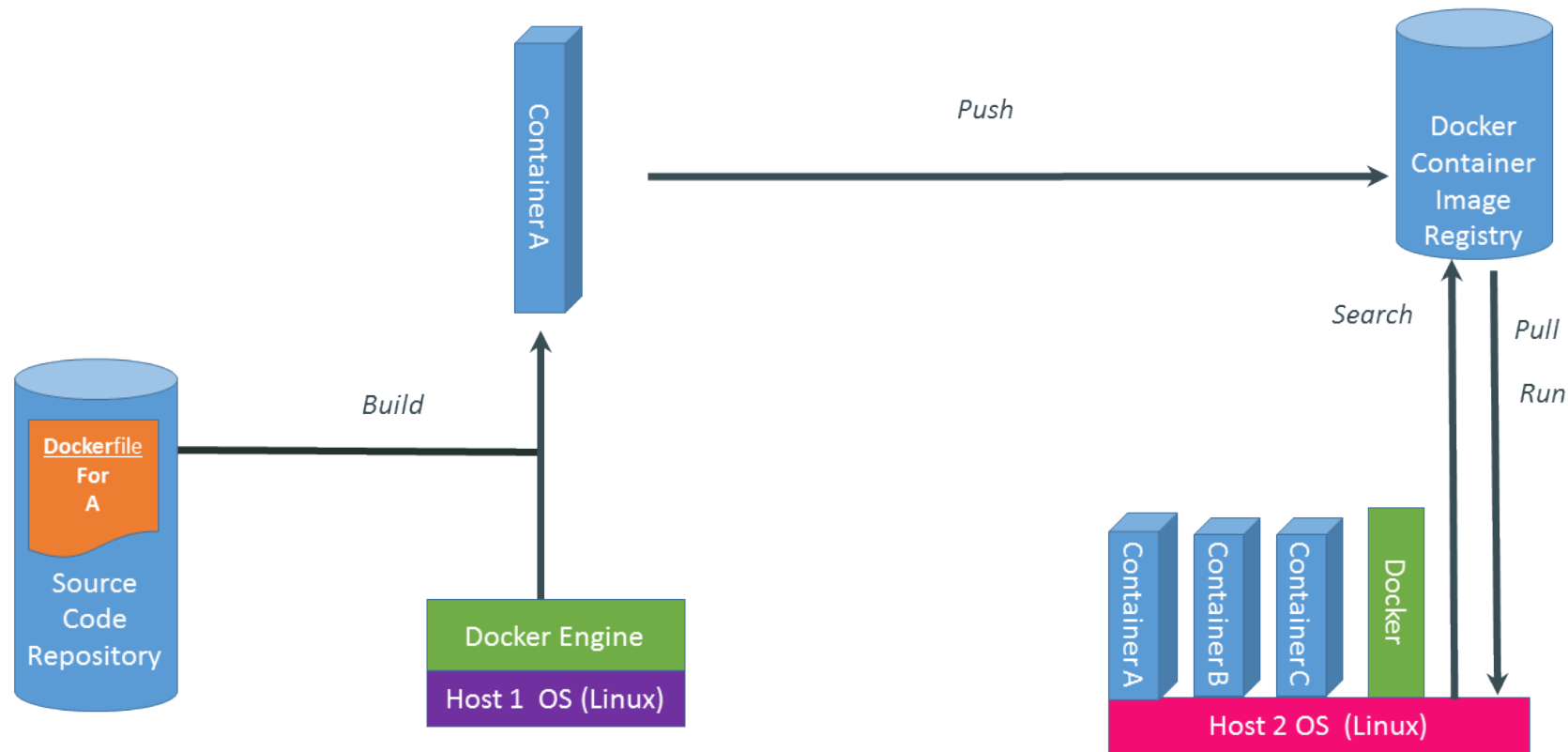
what about  
Windows?

shut up!

# OSX?

- Virtualbox + Vagrant
- [github.com/steeve/boot2docker](https://github.com/steeve/boot2docker)
- Docker OSX native client ( $\geq 0.7.3$ )

# Typical Workflow



# Demo

[github.com/colinsurprenant/devopsmtl-docker](https://github.com/colinsurprenant/devopsmtl-docker)

```
vagrant@vee wee-ubuntu-13: ~ (ssh)
vagrant@vee wee-ubuntu-13: ~/devopsmtl-docker/images/jruby$ docker build -t devopsmtl/jruby:1.7.9 .
Uploading context 10.24 kB
Uploading context
Step 1 : FROM racker/precise-with-updates:latest
----> e60a8d0d6464
Step 2 : MAINTAINER Colin Surprenant <colin.surprenant@gmail.com>
----> Using cache
----> 9b696d0ba117
Step 3 : ENV DEBIAN_FRONTEND noninteractive
----> Using cache
----> 9bce453f8d21
Step 4 : RUN apt-get update; apt-get upgrade -y; apt-get install -y git-core wget curl build-essential zlib1g-dev libssl-dev libreadline6-
dev libyaml-dev libgdbm-dev libffi-dev libxml2-dev libxslt1-dev libncurses5-dev libfuse2
----> Using cache
----> 27643733e403
Step 5 : RUN (cd /tmp ; apt-get download fuse; dpkg-deb -x fuse_* .; dpkg-deb -e fuse_*; rm fuse_*.deb; echo -en '#!/bin/bash\nexit 0\n' >
DEBIAN/postinst; dpkg-deb -b . /fuse.deb; dpkg -i /fuse.deb)
----> Using cache
----> 6ca45c36cbea
Step 6 : RUN apt-get install -y openjdk-7-jdk ant
----> Using cache
----> ff303f9c6ed3
Step 7 : ENV JAVA_HOME /usr/lib/jvm/java-7-openjdk-amd64
----> Using cache
----> ff7ee026d9c0
Step 8 : RUN git clone https://github.com/ctophenson/rhenv /usr/local/rhenv; mkdir -p /usr/local/rhenv/plugins
```

# Further Topics

- Deployment, orchestration, discovery
  - Container linking
  - Ambassador Container
    - [docs.docker.io/en/latest/use/ambassador\\_pattern\\_linking/](https://docs.docker.io/en/latest/use/ambassador_pattern_linking/)
  - CoreOS
  - Shipyard
- PAAS
  - Dokku, Deis, Cocaine, Flynn
- Security