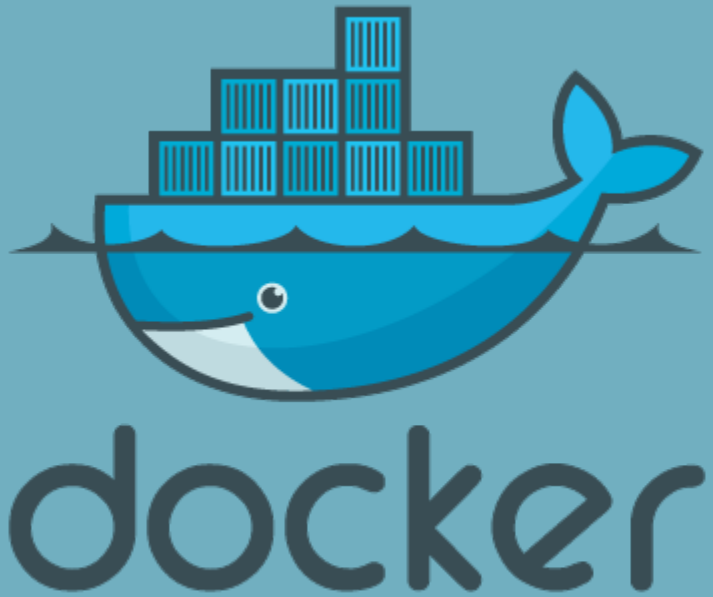


DevOps Montréal

Jan 6, 2014

Colin Surprenant

@colinsurprenant

github.com/colinsurprenant/devopsmtl-docker

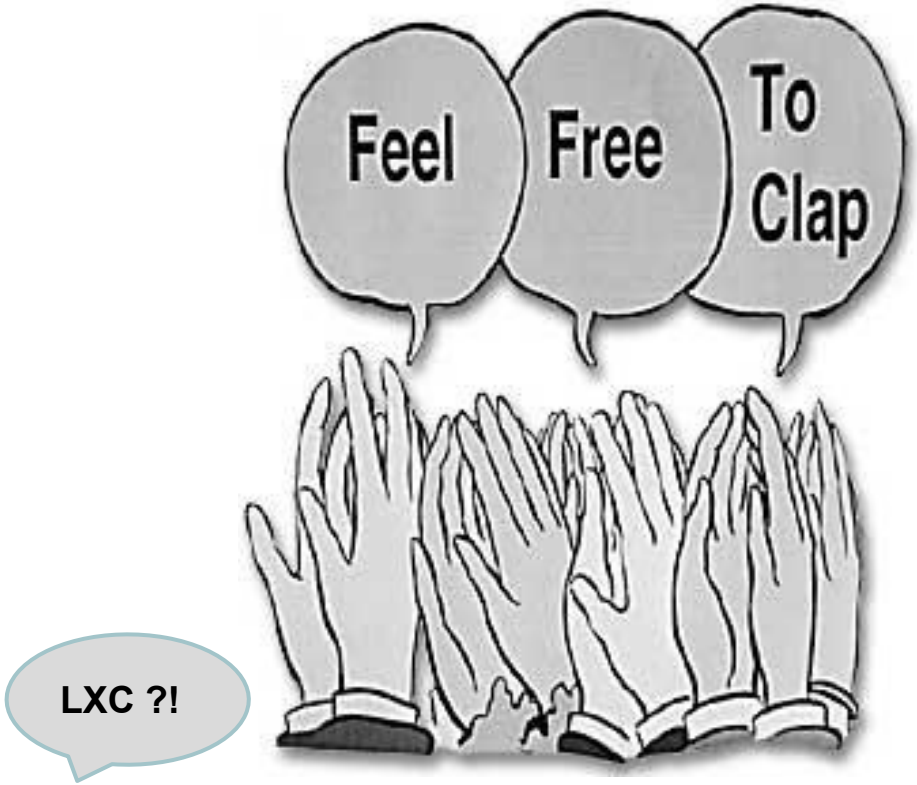
What is Docker?

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!



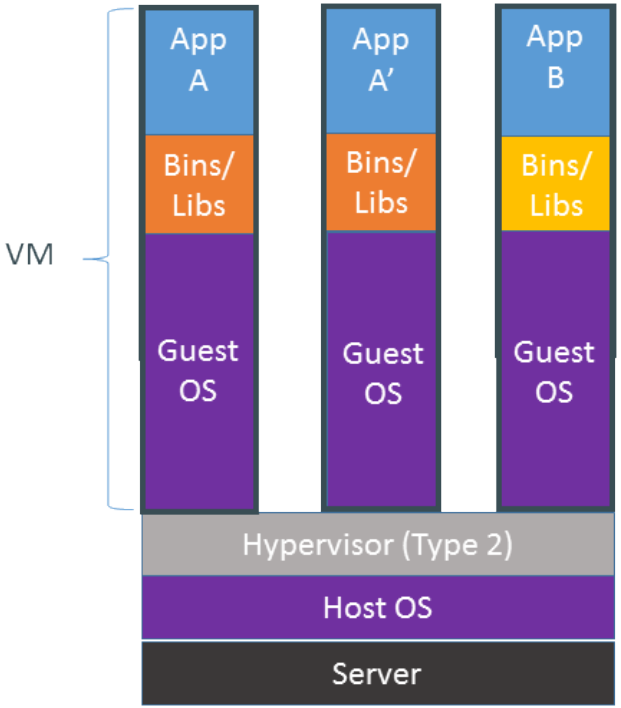
What is LXC?

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

LXC

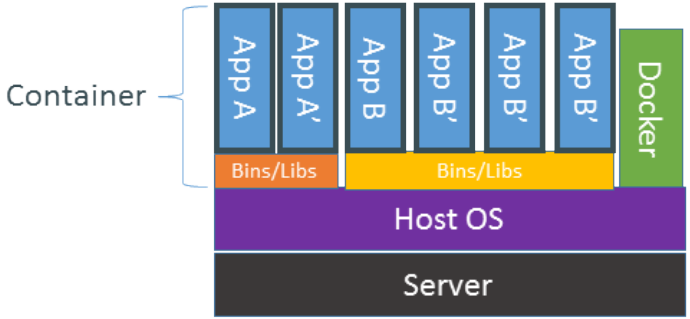
- Run Linux within Linux
- Lightweight VM
  - own process space
  - own network interface
  - SHARE kernel with host
- A container is a group of isolated processes
  - cgroups
  - namespace
- “chroot” on steroid

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

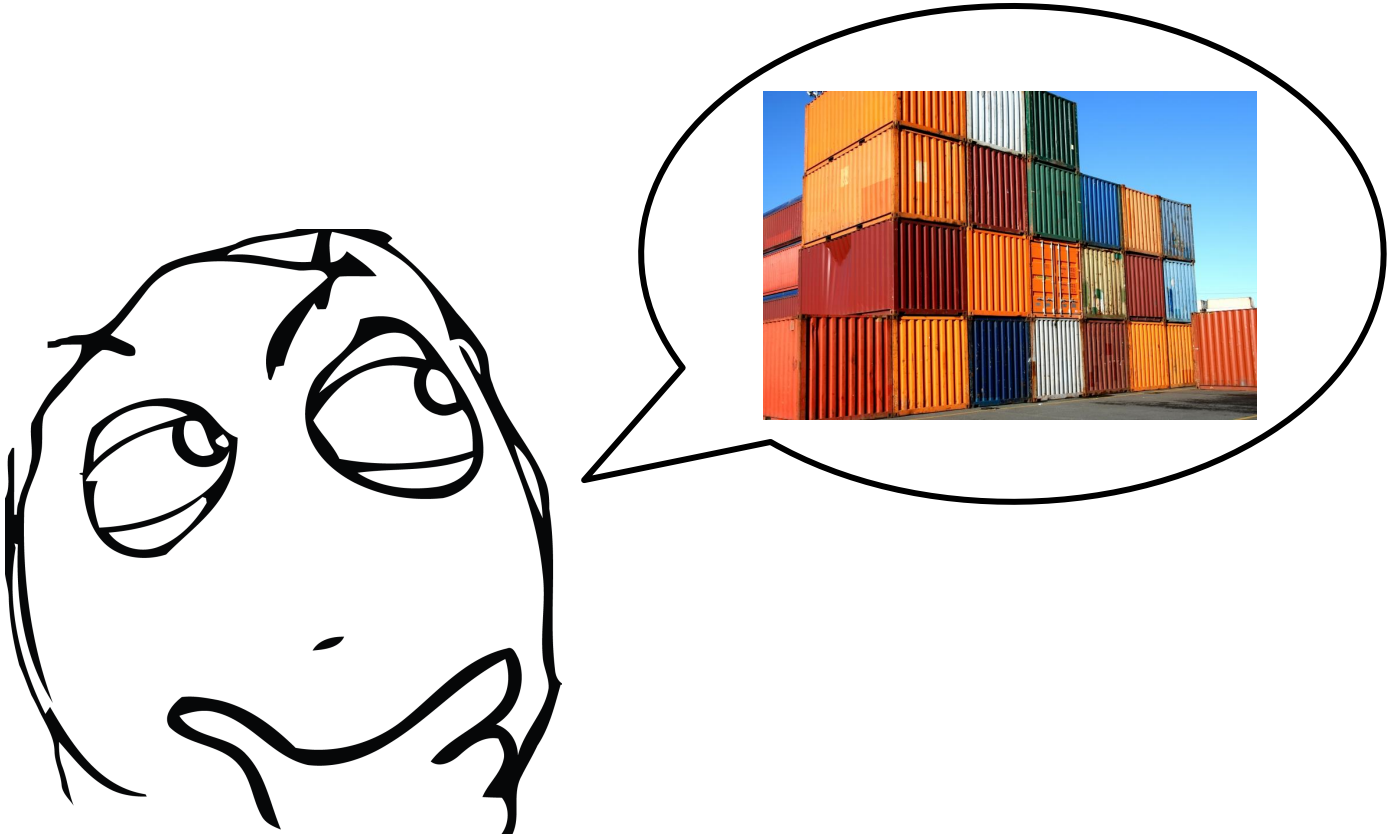


LXC Performance?

Negligible overhead

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

So then, what is Docker?

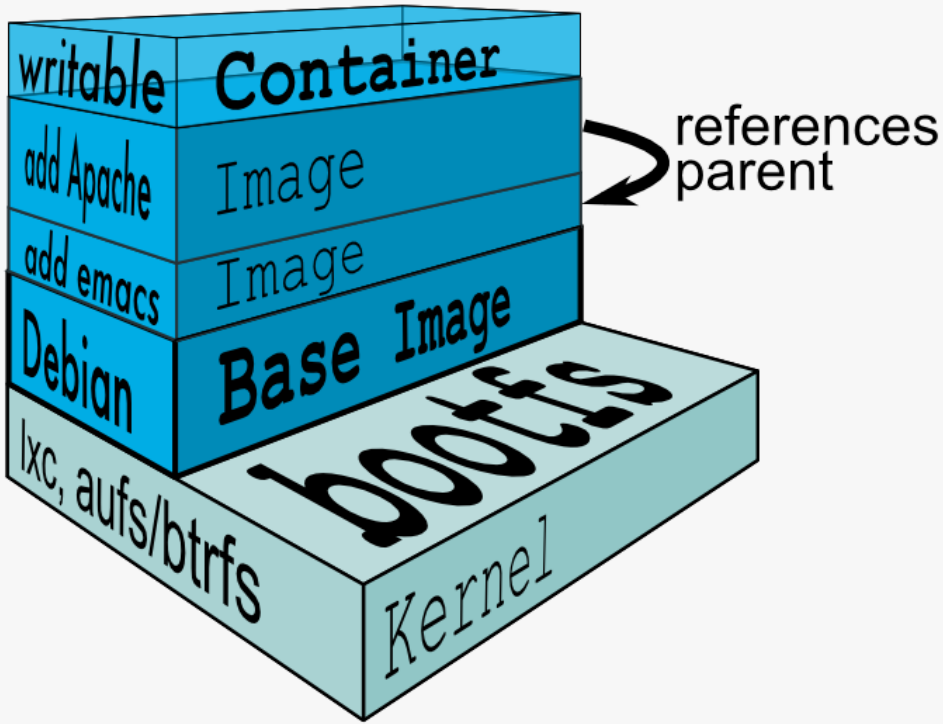


Docker

- Tools to easily build images
- Share images using repositories
  - public repository: index.docker.io
  - 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(™)
- 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

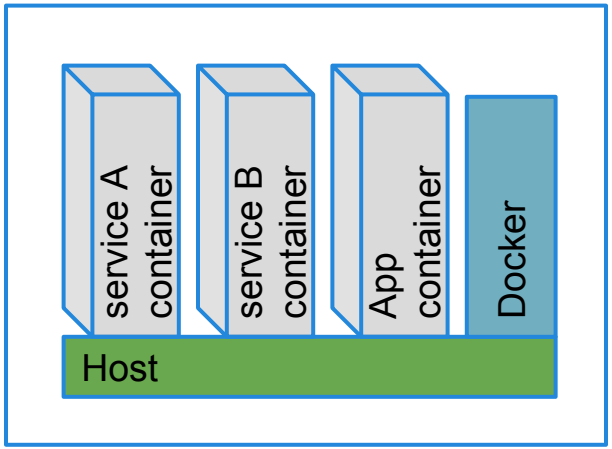
- my code
- my libraries
- my package manager
- my app
- my data

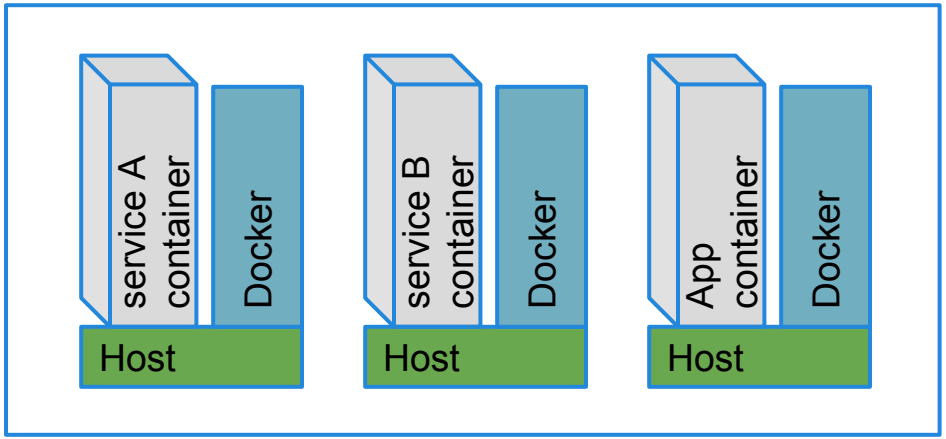
DevOps

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

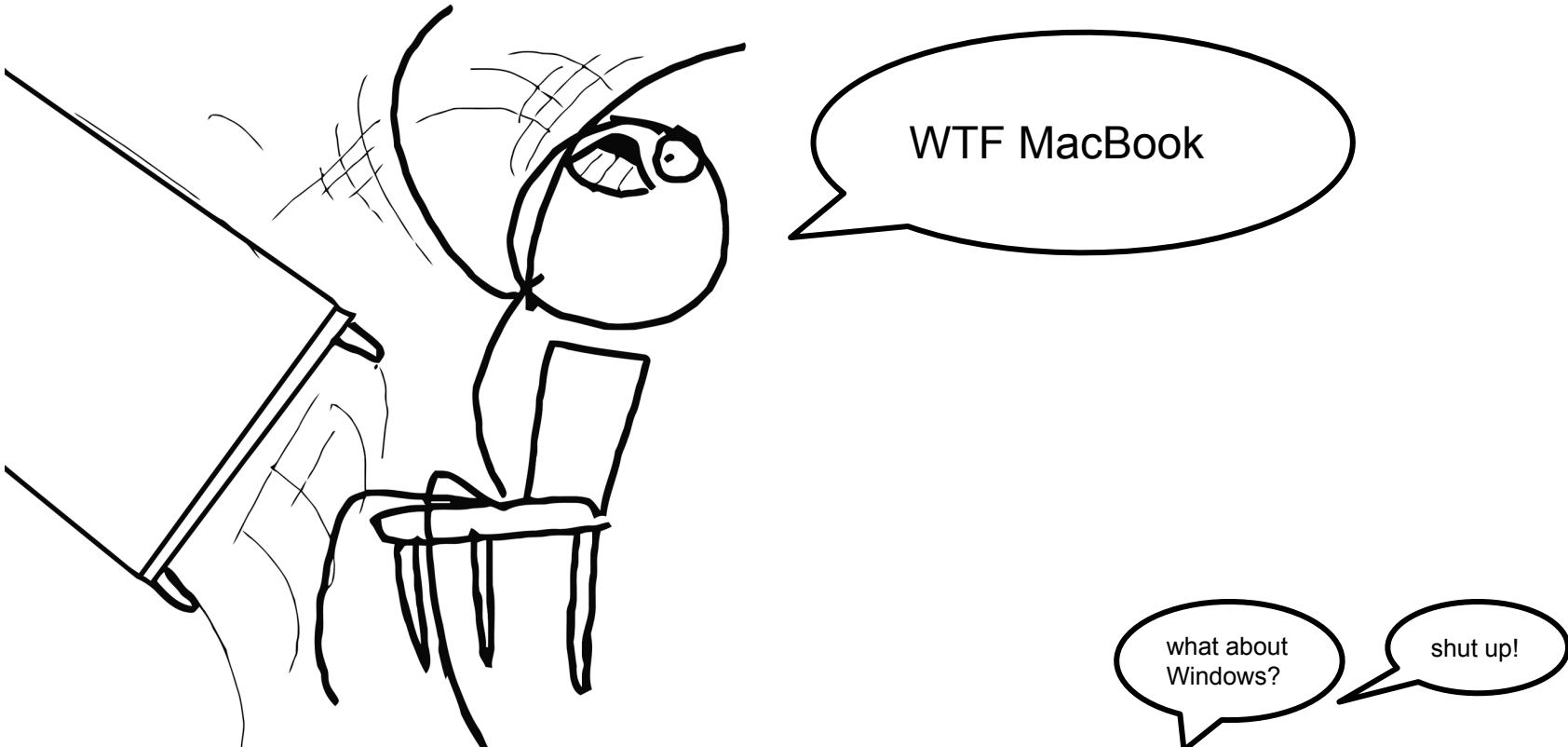
- Path to *The Twelve-Factors App*

Requirements

- Linux kernel >= 3.8
- AUFS
- LXC
- 64 bits

- Recommended
  - Ubuntu 12.04 with upgraded kernel
  - Ubuntu 13.04

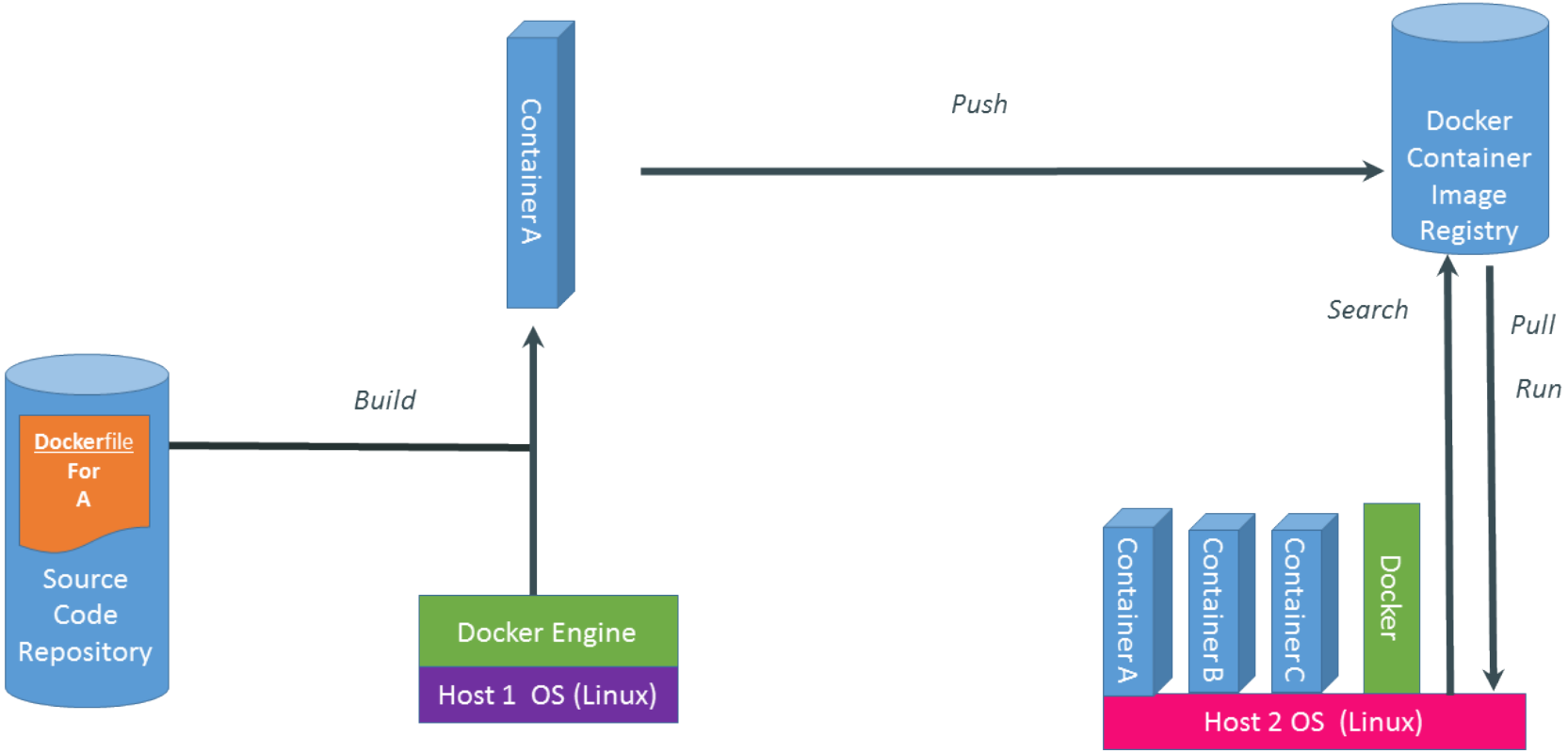
Bro, I’m developing on a MacBook



OSX?

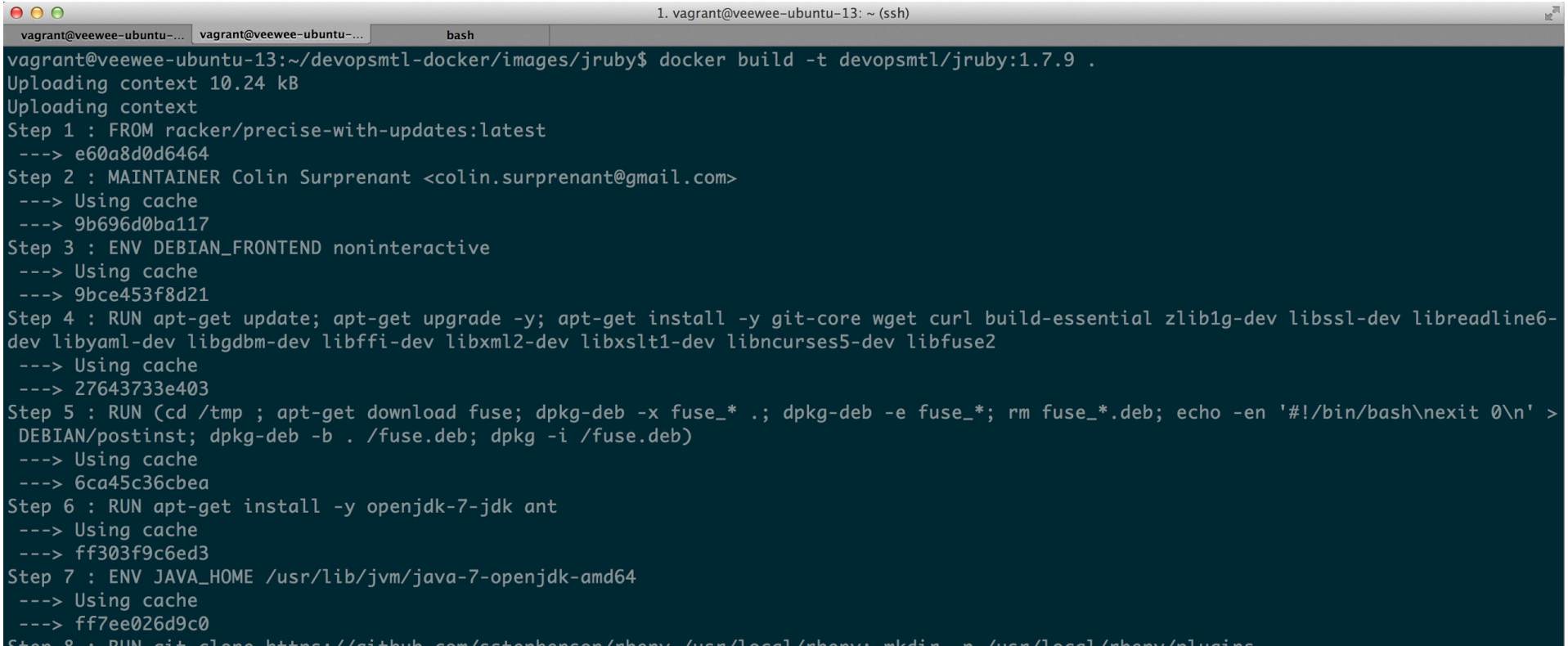
- Virtualbox + Vagrant
- github.com/steeve/boot2docker
- Docker OSX native client (>= 0.7.3)

Typical Workflow



Demo

github.com/colinsurprenant/devopsmtl-docker



Further Topics

- Deployment, orchestration, discovery
  - Container linking
  - Ambassador Container
    - docs.docker.io/en/latest/use/ambassador_pattern_linking/
  - CoreOS
  - Shipyard
- PAAS
  - Dokku, Deis, Cocaine, Flynn
- Security