

Learn You a Docker for Great Good

(Apologies to Haskell)

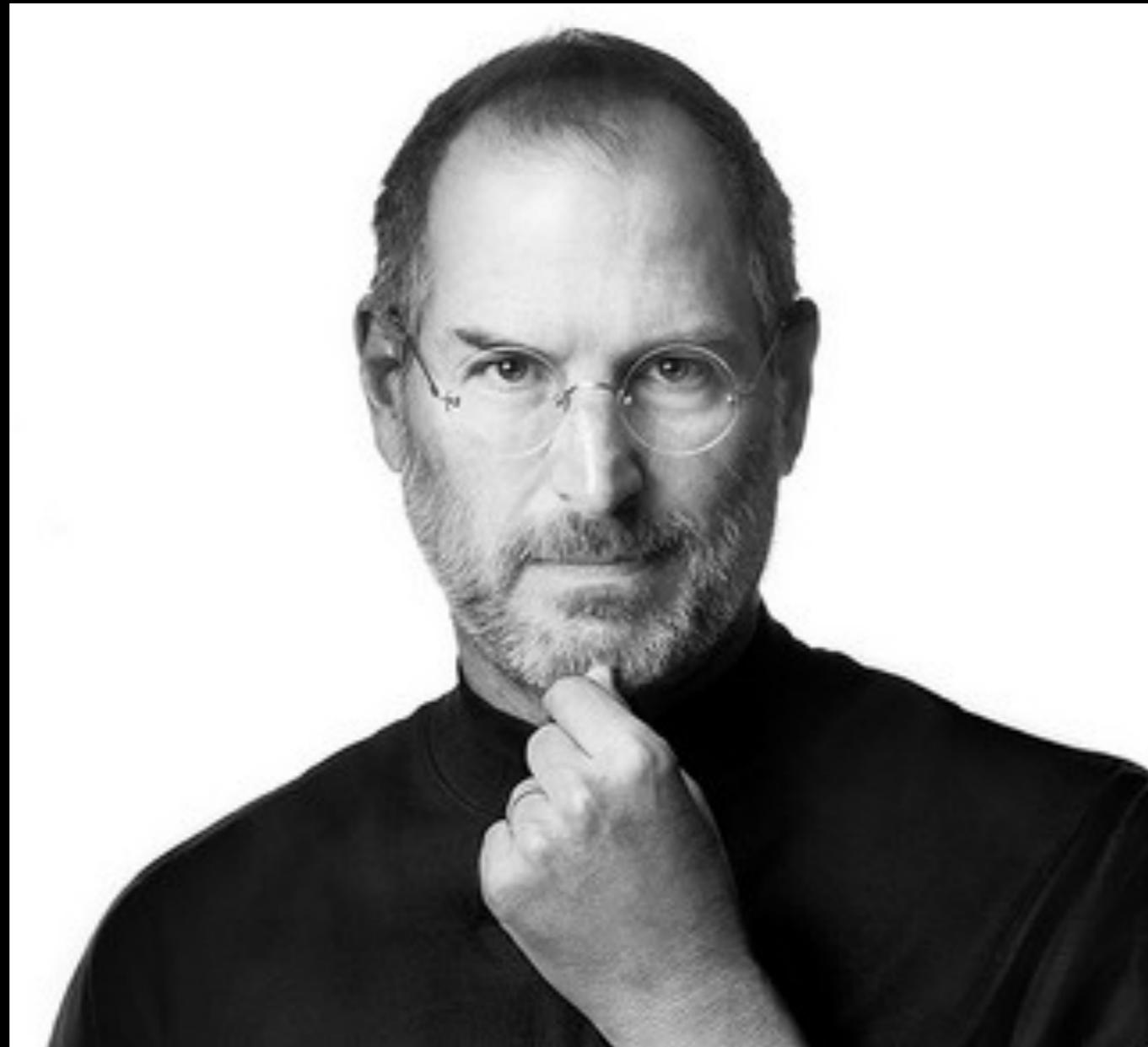
A Brown Bag presentation by Arthur Ketcham at Navigating Cancer. March 2016



docker

What is Docker?

And what's the deal with whales and containers?

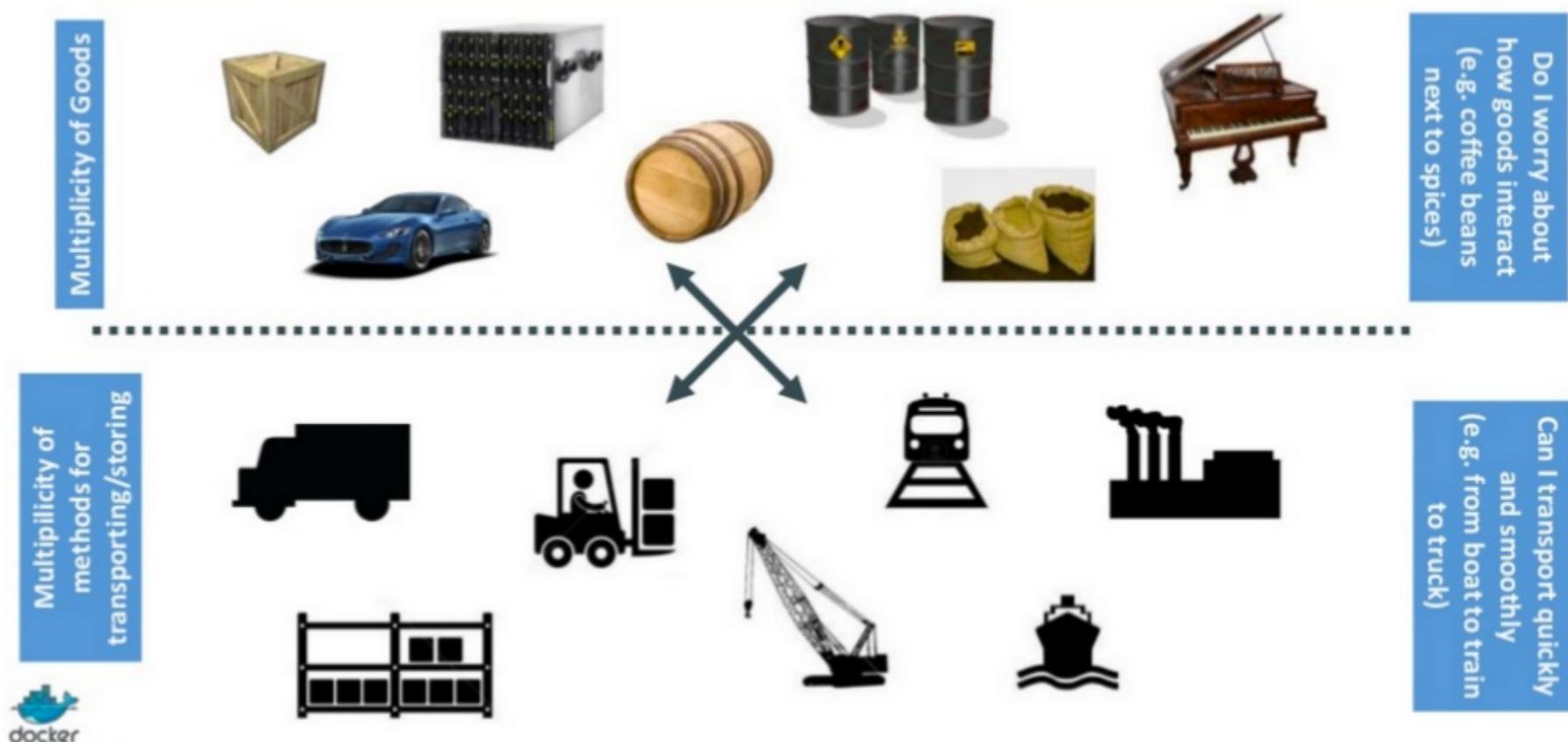


“Real artists ship”

-- Steve Jobs

“Ship” you say?

Cargo Transport Pre-1960



(Courtesy <http://www.slideshare.net/dotCloud/why-docker>)

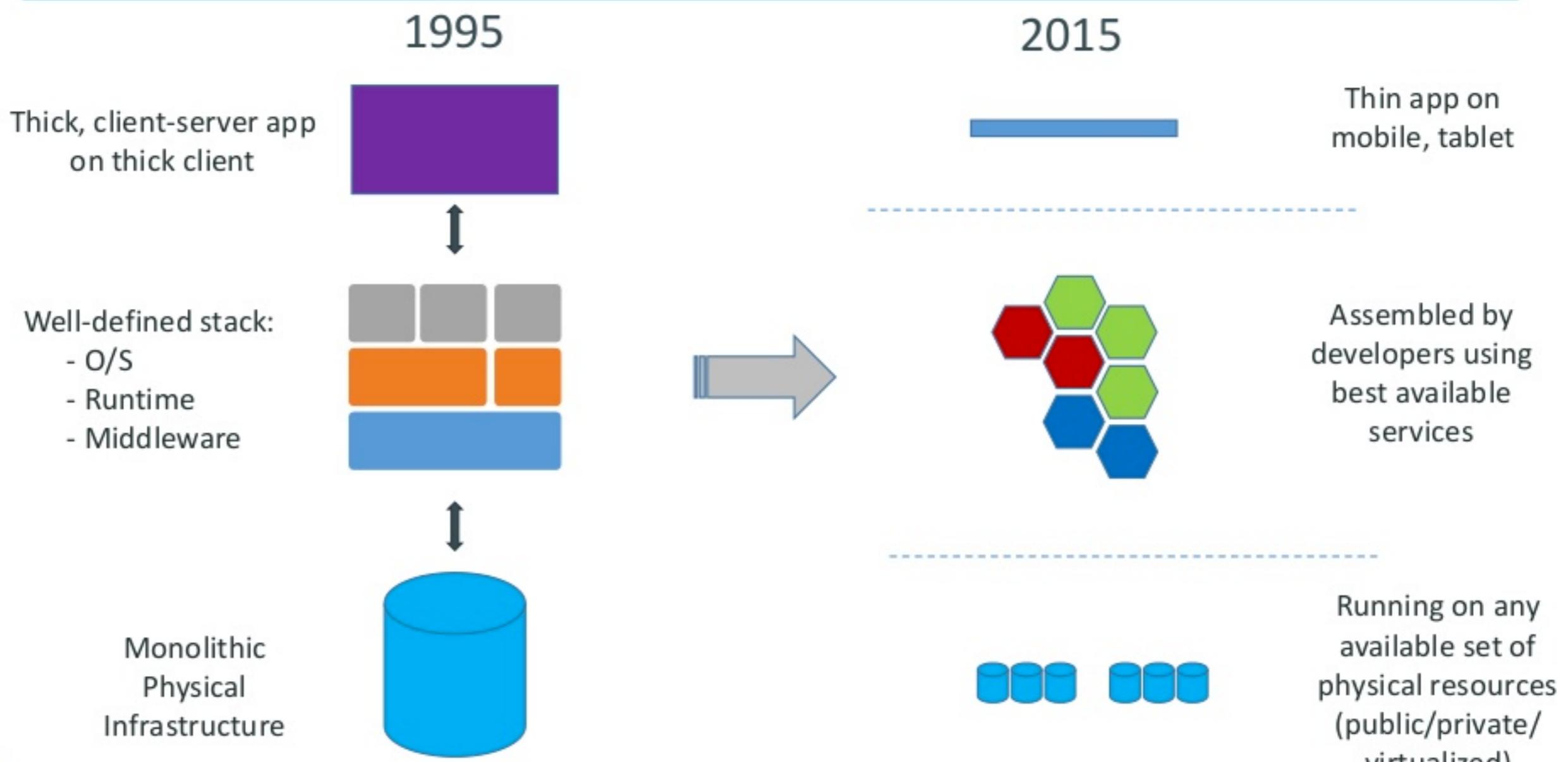
Solution: Intermodal Shipping Container



Modern Shipping



Market View: Evolution of IT



The Challenge

Multiplicity of Stacks



Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2



Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs



User DB

postgresql + pgv8 + v8



Queue

Redis + redis-sentinel



Analytics DB

hadoop + hive + thrift + OpenJDK



Web frontend

Ruby + Rails + sass + Unicorn



API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Do services and apps interact appropriately?

Multiplicity of hardware environments



Development VM



QA server

Customer Data Center



Public Cloud



Production Cluster



Disaster recovery

Contributor's laptop



Can I migrate smoothly and quickly?

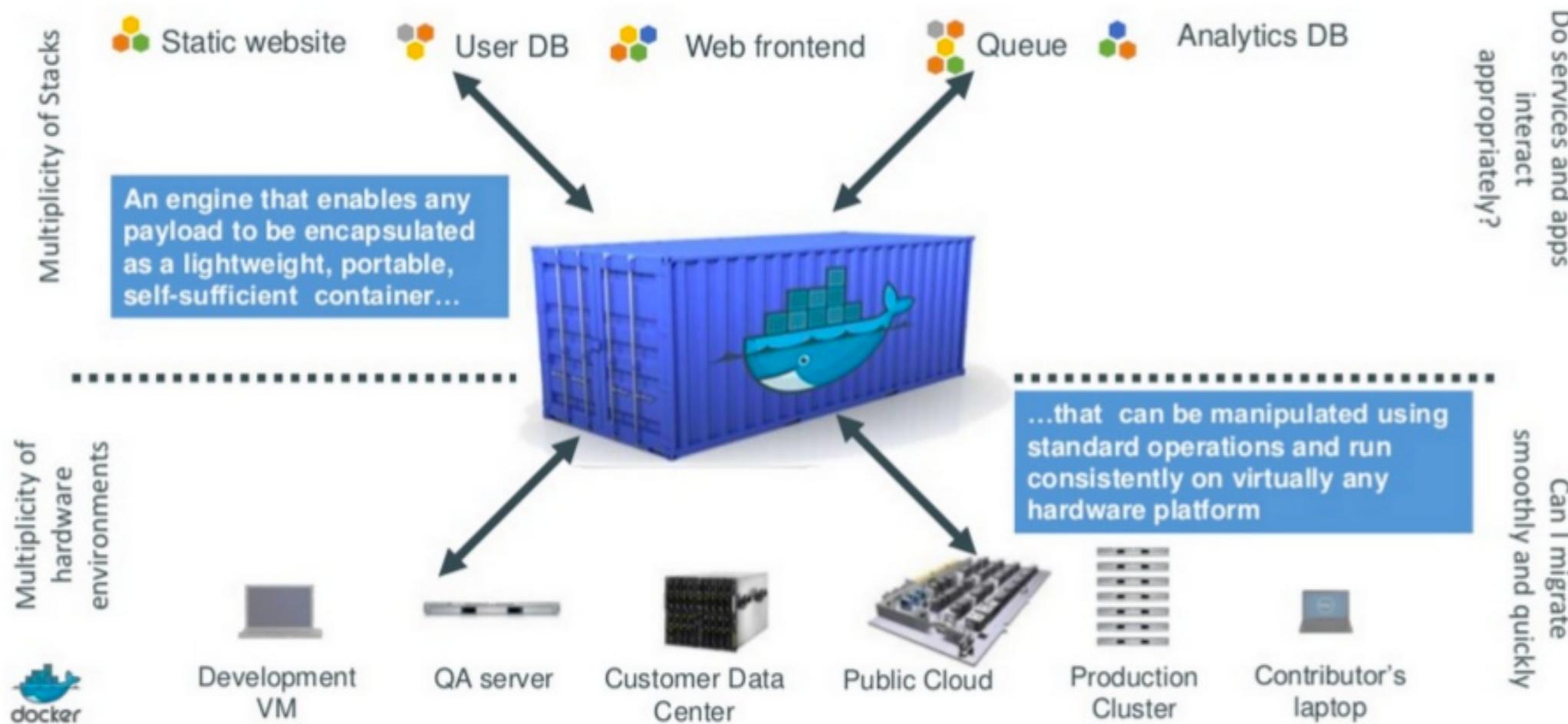
docker

Results in N X N compatibility nightmare

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



Docker is a shipping container system for code



Docker's Forerunners

- Chroot Jails
- Virtual Machines: Xen, VMWare, etc.
- OpenVZ
- Libvirt: Tools and API for Virtualization
- KVM: Kernel-based Virtual Machine
- LXC: Linux Containers

Docker is an LXC Framework

Why Linux Containers (LXC)



- **Fast**
 - Runtime performance near bare metal speeds
 - Management operations (run, stop , start, etc.) in seconds / milliseconds
- **Agile**
 - VM-like agility – it's still “virtualization”
 - Seamlessly “migrate” between virtual and bare metal environments
- **Flexible**
 - Containerize a “system”
 - Containerize “application(s)”
- **Lightweight**
 - Just enough Operating System (JeOS)
 - Minimal per container penalty
- **Inexpensive**
 - Open source – free – lower TCO
 - Supported with out-of-the-box modern Linux kernel
- **Ecosystem**
 - Growing in popularity
 - Vibrant community & numerous 3rd party apps

A Few Quick Concepts

- **Container**: A running isolated "app", like a mini-OS
 - libcontainer interface to kernel
 - Has own root userspace
 - Single process (usually)
 - Ephemeral. Data gone on termination
- **Images**: A frozen, read-only snapshot of a container
 - Composable in differential layers
 - Build your own or Pull from a public registry
 - Publishable

A Few More Concepts

- **Registry**: A public or private image repository service. Official repos FTW.
- **Docker Hub**: Like Github!
- **Dockerfile**: A recipe to build a container, using other images, commands, and local files
- **Build / Run / Commit / Pull / Push**

Let's Try It

We'll run a three-layer build of a simple web app, using

- Debian
- Sinatra
- Redis

A "Redis" Docker File

```
FROM ubuntu:14.04
MAINTAINER James Turnbull "james@example.com"

RUN apt-get -yqq update && apt-get -yqq
install redis-server redis-tools

EXPOSE 6379

ENTRYPOINT ["/usr/bin/redis-server"]
```

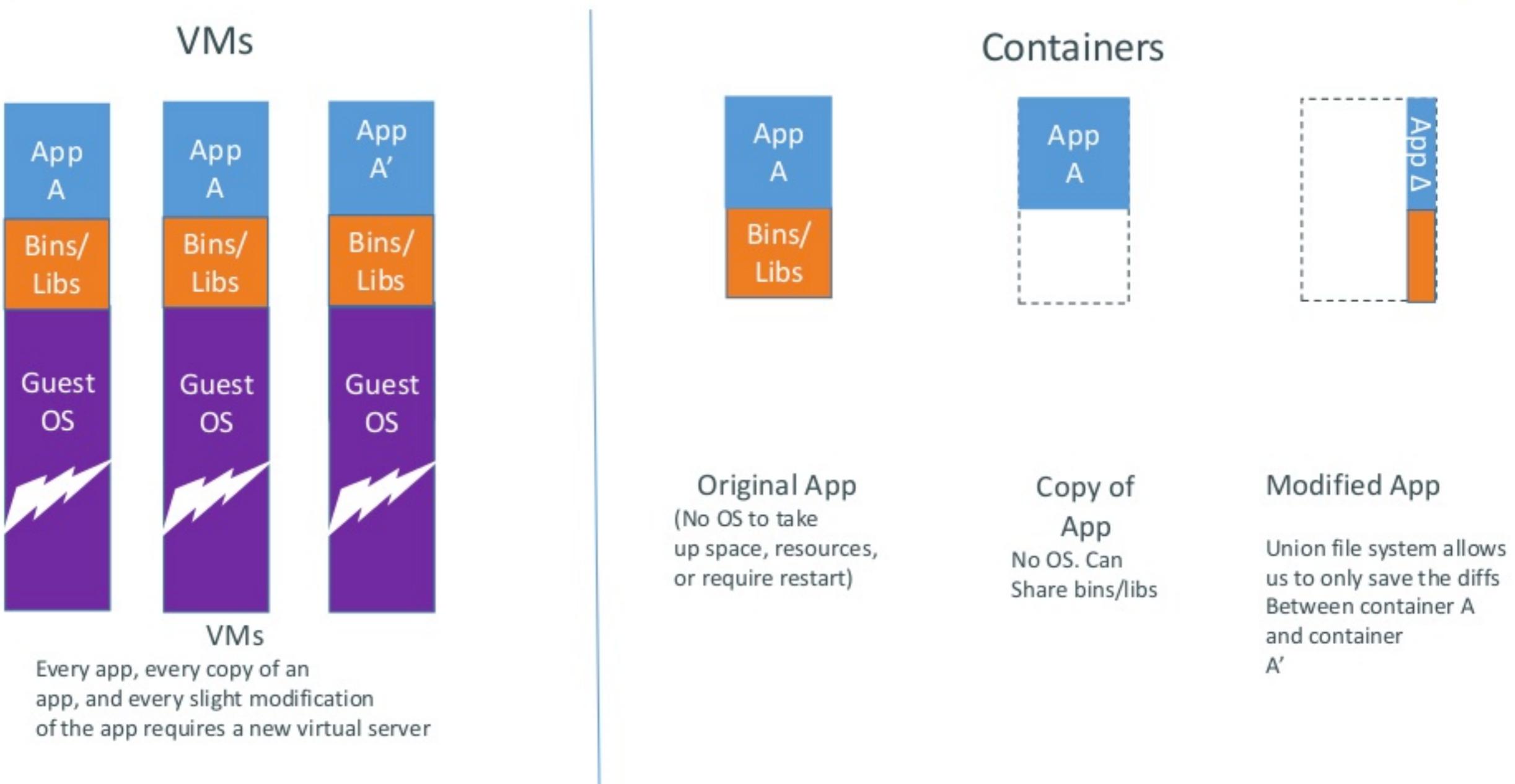
A "Sinatra App" Docker File

```
FROM ruby:2.2-onbuild
MAINTAINER @rmetzler
```

```
ENTRYPOINT ["ruby", "/usr/src/app/web.rb"]
```


Fast, ain't it?

Why are Docker containers lightweight?



Docker Image Size Comparison

ubuntu:latest

188 mb

Layers: 4

busybox:latest

2 mb

Layers: 3

centos:latest

172 mb

Layers: 3

opensuse:latest

82 mb

Layers: 2

alpine:latest

5 mb

Layers: 1

ADD file:c8f078961a543cd...

188 mb

MAINTAINER Jérôme Peta...

0 bytes

MAINTAINER The CentOS ...

0 bytes

MAINTAINER Flavio Castelli

0 bytes

RUN echo '#!/bin/sh' > /usr...

195 kb

ADD file:8cf517d90fe79547...

2 mb

ADD file:82835f82606420c...

172 mb

ADD file:30a527143b57cd1...

82 mb

RUN sed -i 's/^#\!/(deb.*u...

2 kb

CMD "/bin/sh"

0 bytes

CMD "/bin/bash"

0 bytes

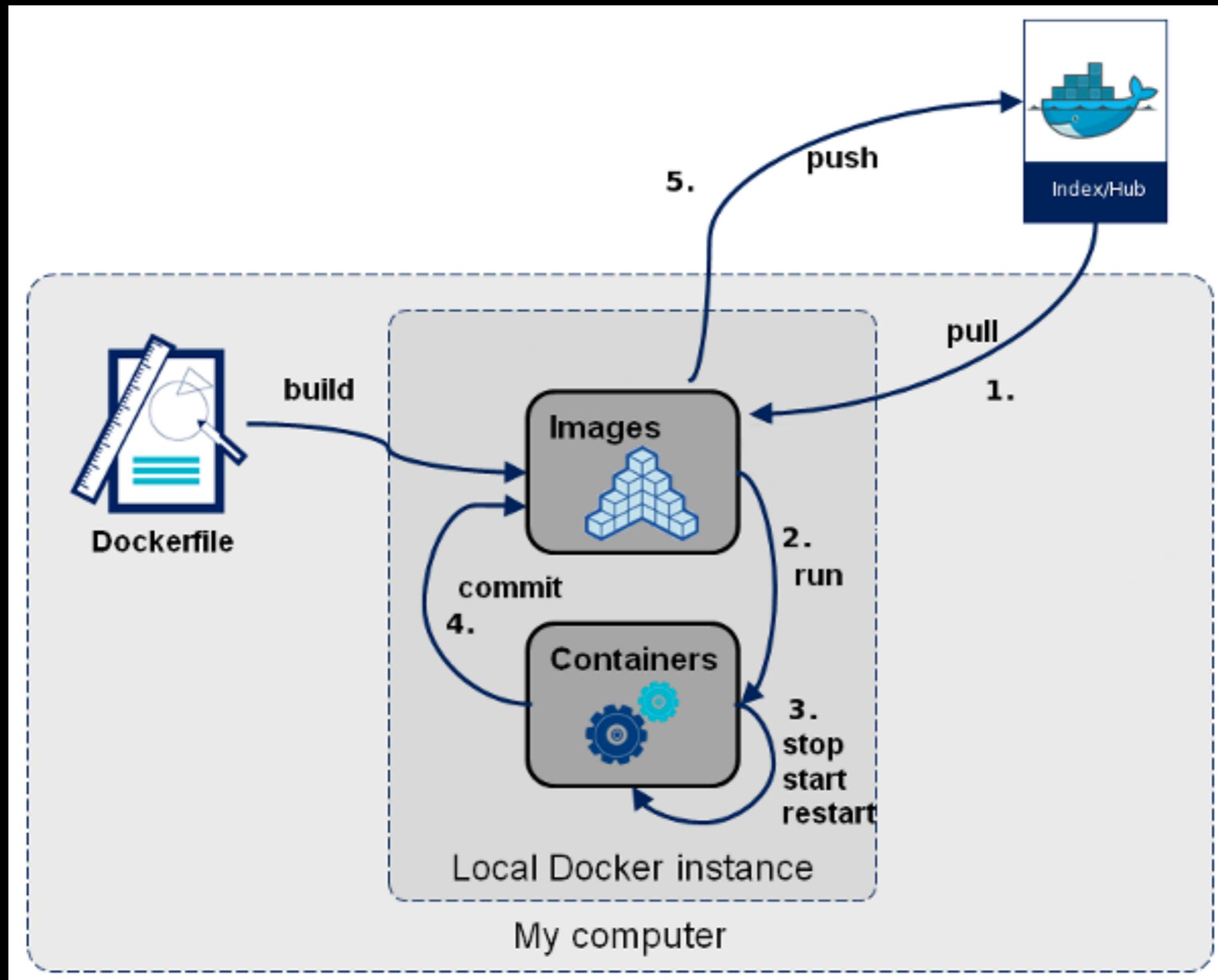
CMD "/bin/bash"

0 bytes

CenturyLink Labs

via <https://www.brianchristner.io/docker-image-base-os-size-comparison/>

Overall Docker Workflow



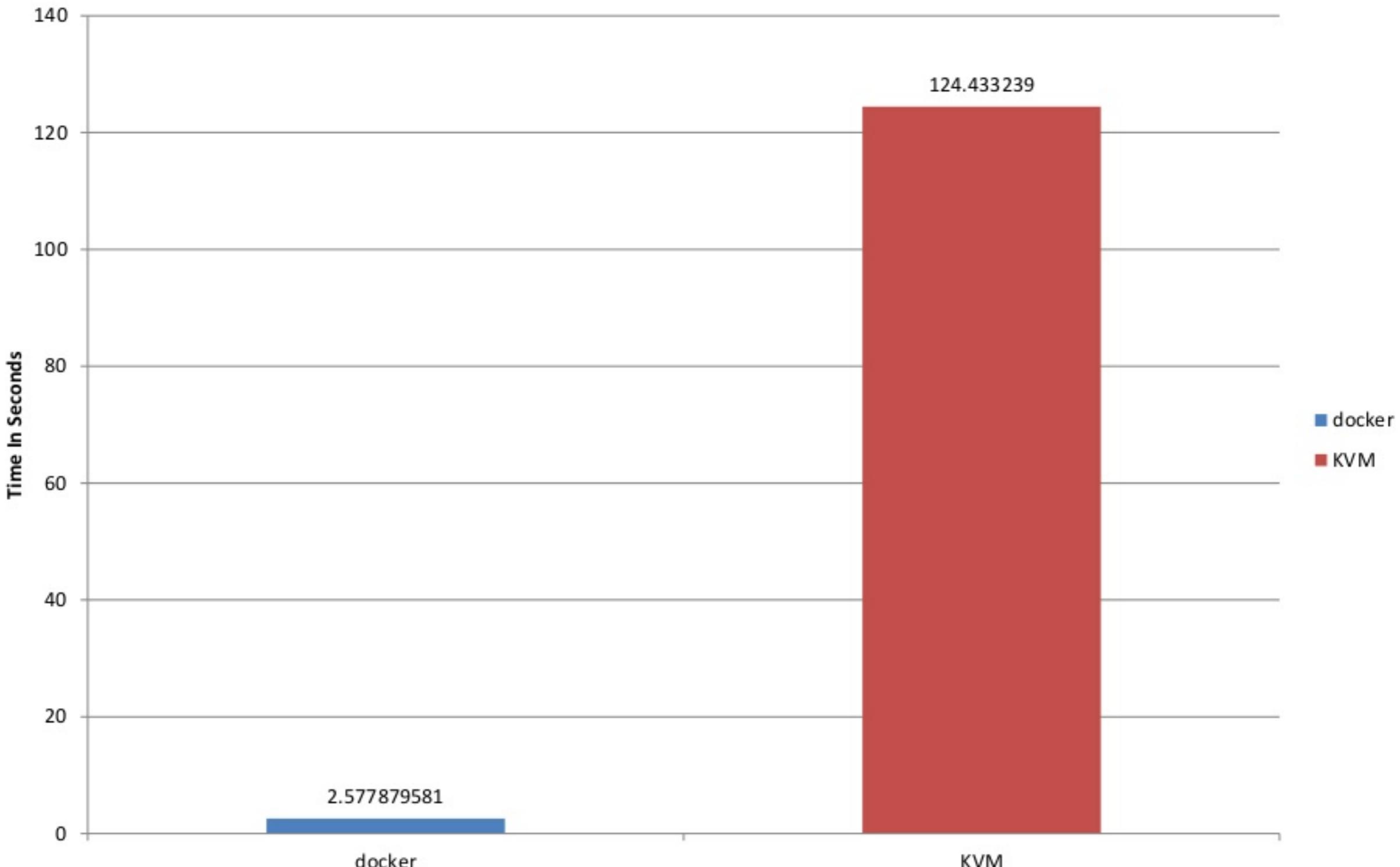
Deployment of Containers

	Ships within ...	Manual deployment takes ...	Automated deployment takes ...	Boots in ...
Bare metal	Days	Hours	Minutes	Minutes
Virtual machine	Minutes	Minutes	Seconds	< Minute
Container	Seconds	Minutes	Seconds	Seconds

Cloudy Performance: Serial VM Reboot

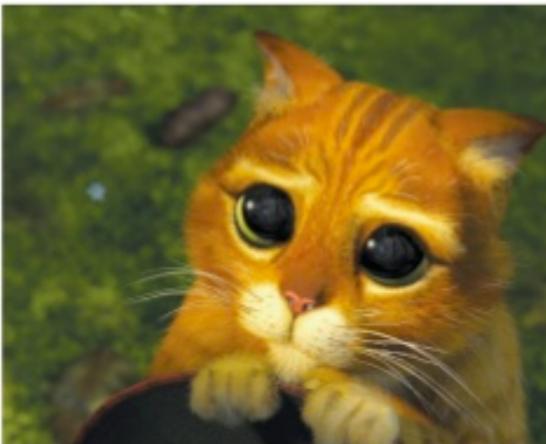


Average Server Reboot Time



Service Model

Borrowed from
@randybias at Cloudscaling
<http://www.slideshare.net/randybias/the-cloud-revolution-cyber-press-forum-philippines>



- Pets are given names like `pussinboots.cern.ch`
- They are unique, lovingly hand raised and cared for
- When they get ill, you nurse them back to health



- Cattle are given numbers like `vm0042.cern.ch`
- They are almost identical to other cattle
- When they get ill, you get another one

- Future application architectures should use Cattle but Pets with strong configuration management are viable and still needed

DevOps Philosophy

- Separate the changing (data) from the immutable (app code).
- Your data are always "pets", but make everything else "cattle".
- Cattle are easily replaced, with less work.
- One "process" per container

What's in it for Developers?

- No worries about dependencies
- Each app runs in isolation, varying versions are no problem.
- Hygienic and portable runtime environment
- Automatable and Scriptable (Great for testing!)
- A VM without the overhead of a VM, instant snapshot, replay, reset

Technical Notes

- Images are stateless and read-only.
- Images can be created from `docker commit` or from Dockerfiles and source code
- Dockerfiles + source is better. Why?
 - Reproducibility & Upgradability
 - Security & Trust
 - Simplicity
 - Tiny

Images and Layers

- Images:
 - Layered in single-inheritance model
 - statically-linked
 - Millions of public images, all extensible

Volumes and Layers

- Volumes:
 - Recall that containers are ephemeral.
 - Use volumes for config and data:
 - `docker run --volume=/host/data:container-data ...`
 - `docker run --env MYVAR2=foo --env-file ./env.list ...`
 - Copy-on-write filesystem model
 - Lots of driver options (aufs, overlay, btrfs, etc.)

WHAT ABOUT DATA?

```
user@host:/$ docker run -d -p 5432:5432 \
    -v `pwd`/data:/var/lib/postgresql \
    -t postgresql
```

"`-v host_dir:container_dir`"

(Courtesy <https://denibertovic.com/talks/supercharge-development-env-using-docker>)

WHAT ABOUT PORTS?

```
user@host:/$ docker run -d -p 5432:5432 -t postgresql  
"-p host_port:container_port"
```

(Courtesy <https://denibertovic.com/talks/supercharge-development-env-using-docker>)

Use Cases—From Our Community

Use Case	Examples	Link
Build your own PaaS	Dokku - Docker powered mini-Heroku. The smallest PaaS implementation you've ever seen	http://bit.ly/191Tgsx
Web Based Environment for Instruction	JiffyLab – web based environment for the instruction, or lightweight use of, Python and UNIX shell	http://bit.ly/12oaj2K
Easy Application Deployment	Deploy Java Apps With Docker = Awesome Running Drupal on Docker Installing Redis on Docker	http://bit.ly/11BCvvu http://bit.ly/15MJS6B http://bit.ly/16EWOKh
Create Secure Sandboxes	Docker makes creating secure sandboxes easier than ever	http://bit.ly/13mZGJH
Create your own SaaS	Memcached as a Service	http://bit.ly/11nL8vh
Automated Application Deployment	Push-button Deployment with Docker	http://bit.ly/1bTKZTo
Continuous Integration and Deployment	Next Generation Continuous Integration & Deployment with dotCloud's Docker and Strider	http://bit.ly/ZwTfoy
Lightweight Desktop Virtualization	Docker Desktop: Your Desktop Over SSH Running Inside Of A Docker Container	http://bit.ly/14RYL6x



Benefits of Linux Containers in the Docker Format



DEPLOYMENT
FLEXIBILITY



OPERATIONAL
EFFICIENCY



SIMPLIFIED
MAINTENANCE



LOWERED
DEPLOYMENT
COSTS

*FIN.
Questions?*