Master in High Performance Computing

An introduction to modern tools for collaborative science

Lecture 4 - Docker and GitHub actions

Luca Heltai <luca.heltai@sissa.it>

Mathematical Analysis, Modeling, and Applications (math.sissa.it)
Theoretical and Scientific Data Science (datascience.sissa.it)

sissa mathLab innovating with mathematics

The Abdus Salam International Centre for Theoretical Physics

SISSA

Introduction to Docker

Adapted from official docker.io slides of November, 2013

Luca Heltai <luca.heltai@sissa.it>

The Challenge

Multiplicity of Stacks

Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

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

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

Public Cloud

Production Cluster

Disaster recovery

Customer Data Center

Contributor's laptop

Production Servers

Can I migrate smoothly and quickly?

docker

The Matrix From Hell

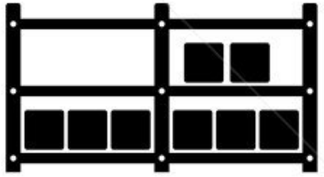
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Cargo Transport Pre-1960

Multiplicity of Goods

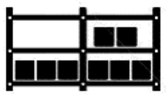
Do I worry about how goods interact (e.g. coffee beans next to spices)

Multipilicity of methods for transporting/storing



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

Also a matrix from hell

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

Solution: Intermodal Shipping Container

Multiplicity of Goods





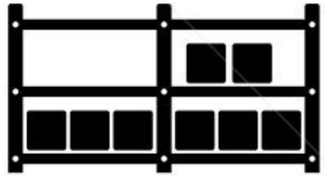


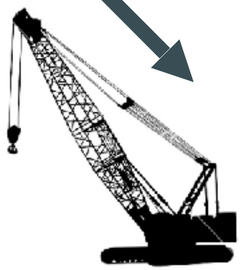
Do I worry about how goods interact (e.g. coffee beans next to spices)

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

Multiplicity of methods for transporting/storing

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

docker

Docker is a shipping container system for code

Multiplicity of Stacks

Static website

User DB

Web frontend

Queue

Analytics DB

Do services and apps interact appropriately?

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container…

…that can be manipulated using standard operations and run consistently on virtually any hardware platform

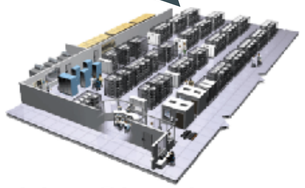
Can I migrate smoothly and quickly

Multiplicity of hardware environments

Development VM

QA server

Customer Data Center



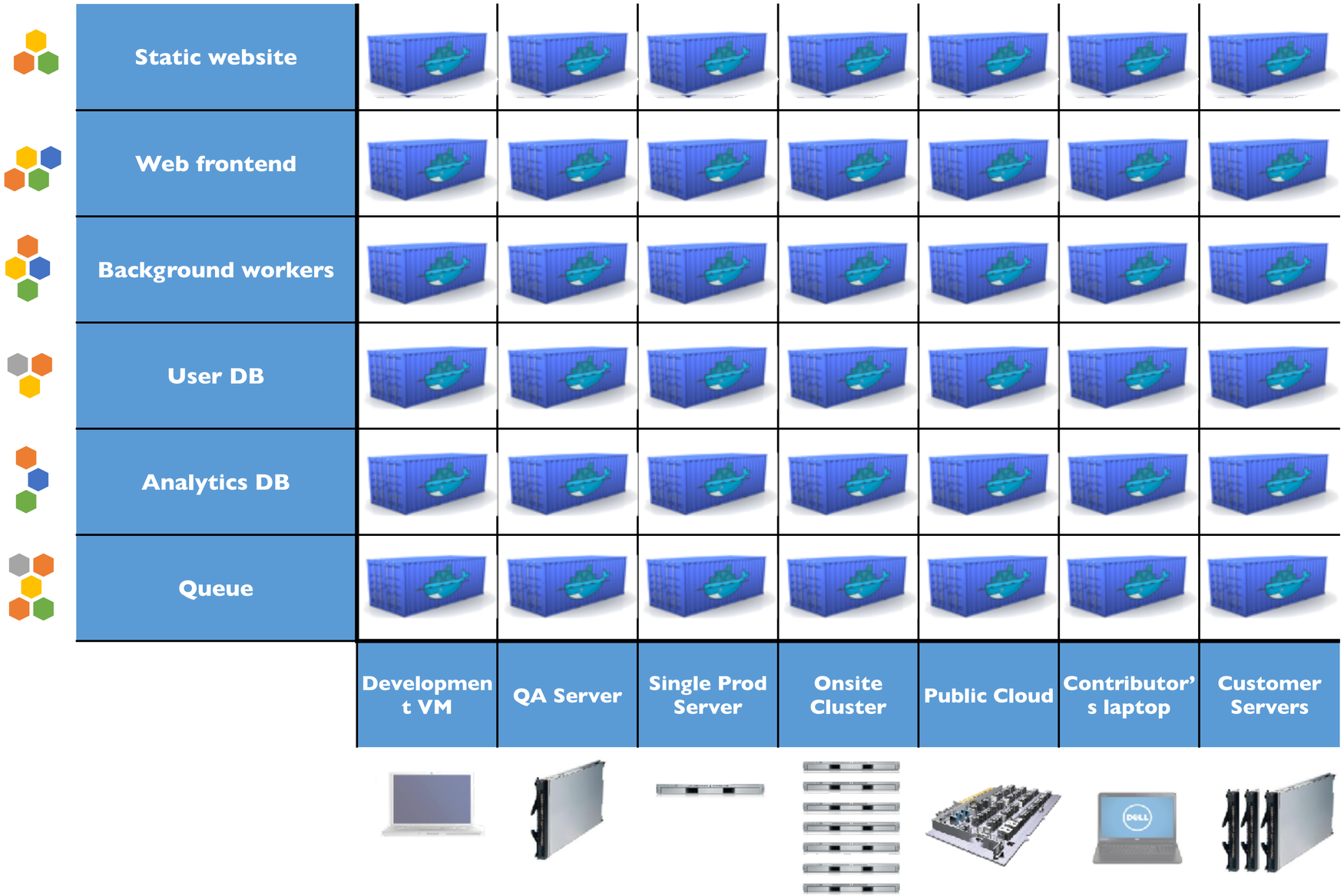
Public Cloud

Production Cluster

Contributor's laptop

docker

Docker eliminates the matrix from Hell



Why Developers Care

- Build once…(finally) run anywhere*
  - A clean, safe, hygienic and portable runtime environment for your app.
  - No worries about missing dependencies, packages and other pain points during subsequent deployments.
  - Run each app in its own isolated container, so you can run various versions of libraries and other dependencies for each app without worrying
  - Automate testing, integration, packaging…anything you can script
  - Reduce/eliminate concerns about compatibility on different platforms, either your own or your customers.
  - Cheap, zero-penalty containers to deploy services? A VM without the overhead of a VM? Instant replay and reset of image snapshots? That's the power of Docker

* With the 0.7 release, we support any x86 server running a modern Linux kernel (3.2+ generally. 2.6.32+ for RHEL 6.5+, Fedora, & related)

docker

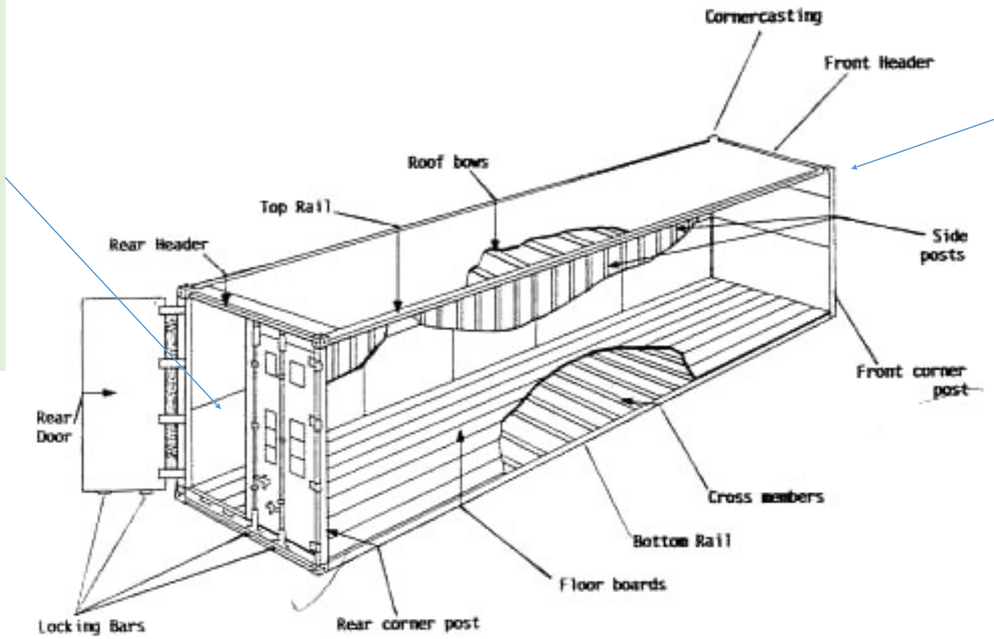
Why Devops Cares?

- Configure once…run anything
  - Make the entire lifecycle more efficient, consistent, and repeatable
  - Increase the quality of code produced by developers.
  - Eliminate inconsistencies between development, test, production, and customer environments
  - Support segregation of duties
  - Significantly improves the speed and reliability of continuous deployment and continuous integration systems
  - Because the containers are so lightweight, address significant performance, costs, deployment, and portability issues normally associated with VMs

docker

Why it works—separation of concerns

- Dan the Developer
  - Worries about what's "inside" the container
    - His code
    - His Libraries
    - His Package Manager
    - His Apps
    - His Data
  - All Linux servers look the same



Major components of the container:

- Oscar the Ops Guy
  - Worries about what's "outside" the container
    - Logging
    - Remote access
    - Monitoring
    - Network config
  - All containers start, stop, copy, attach, migrate, etc. the same way

More technical explanation

WHY

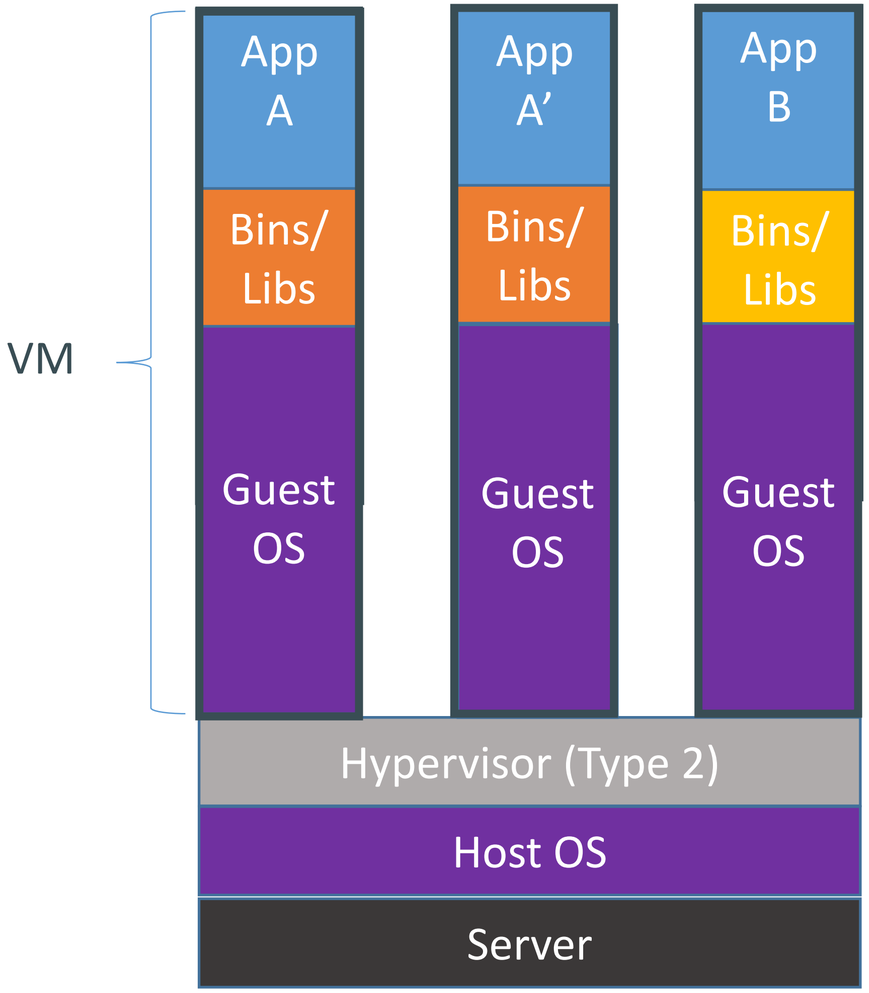
- Run everywhere
  - Regardless of kernel version (2.6.32+)
  - Regardless of host distro
  - Physical or virtual, cloud or not
  - Container and host architecture must match*
- Run anything
  - If it can run on the host, it can run in the container
  - i.e. if it can run on a Linux kernel, it can run

WHAT

- High Level—It's a lightweight VM
  - Own process space
  - Own network interface
  - Can run stuff as root
  - Can have its own /sbin/init (different from host)
  - <<machine container>>
- Low Level—It's chroot on steroids
  - Can also *not* have its own /sbin/init
  - Container=isolated processes
  - Share kernel with host
  - No device emulation (neither HVM nor PV) from host)
  - <<application container>>

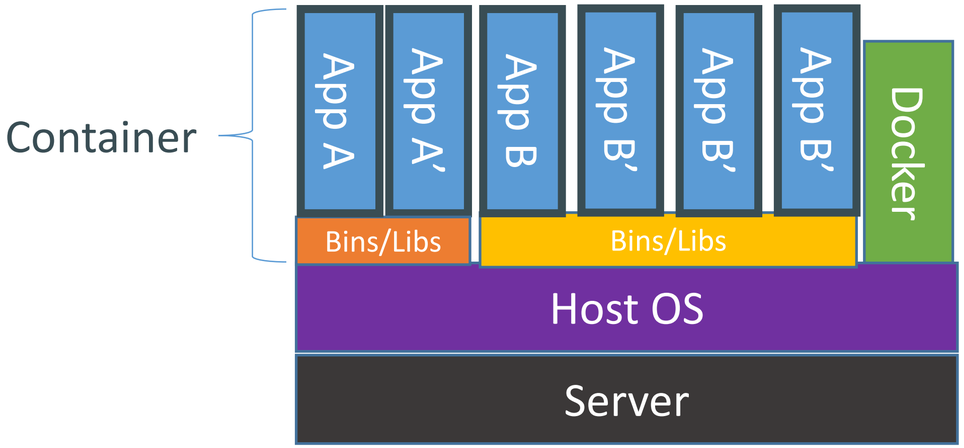
docker

Containers vs. VMs



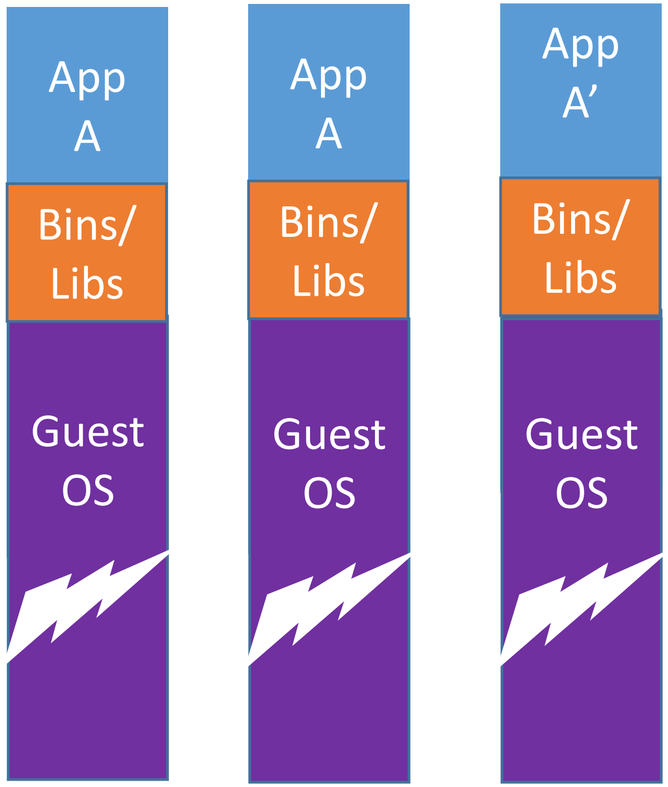
Containers are isolated, but share OS and, where appropriate, bins/libraries

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



Why are Docker containers lightweight?

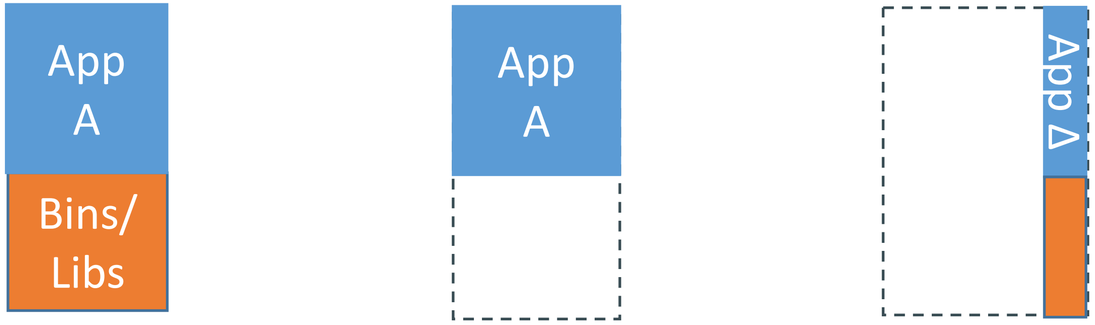
VMs



VMs

Every app, every copy of an app, and every slight modification of the app requires a new virtual server

Containers



Original App

(No OS to take up space, resources, or require restart)

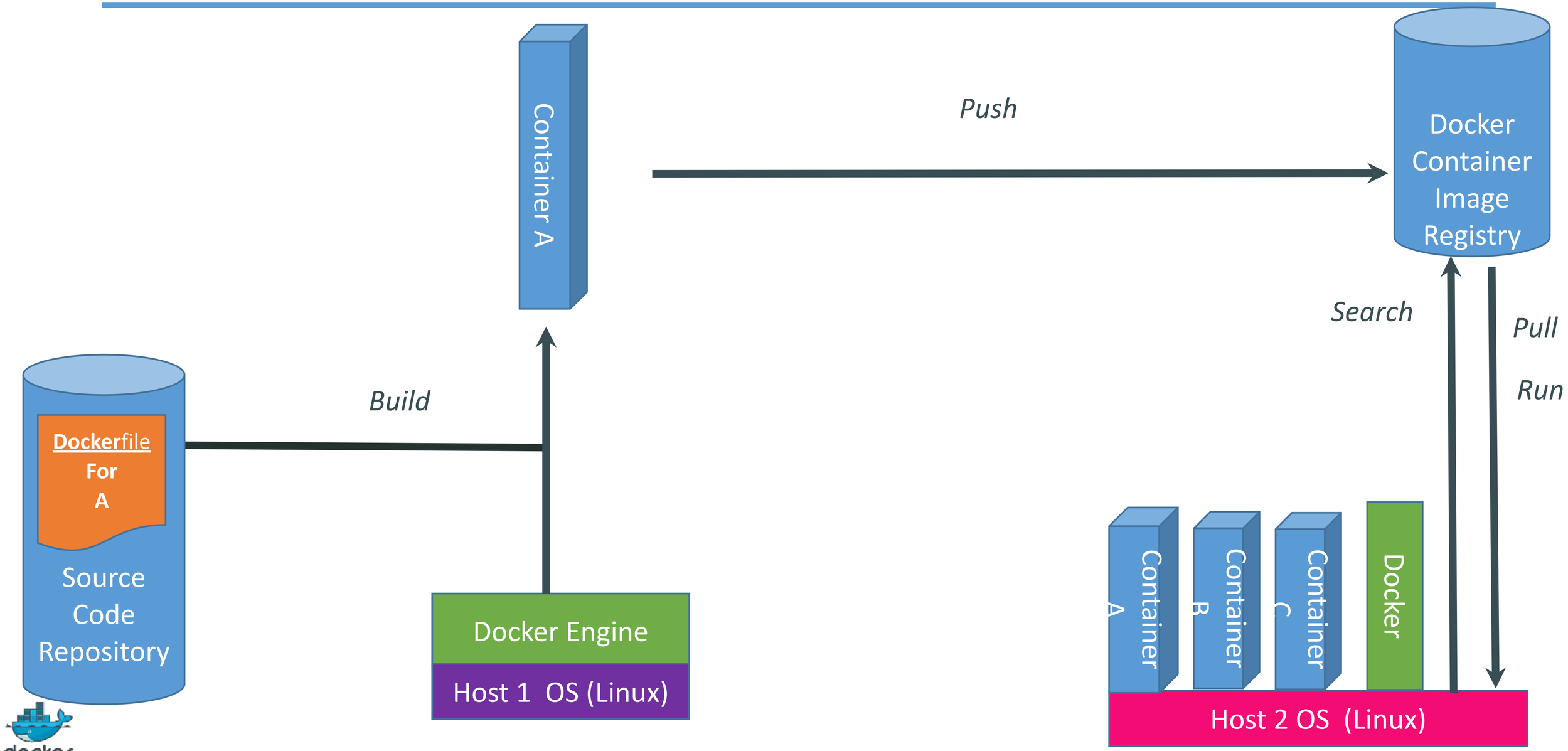
Copy of App

No OS. Can Share bins/libs

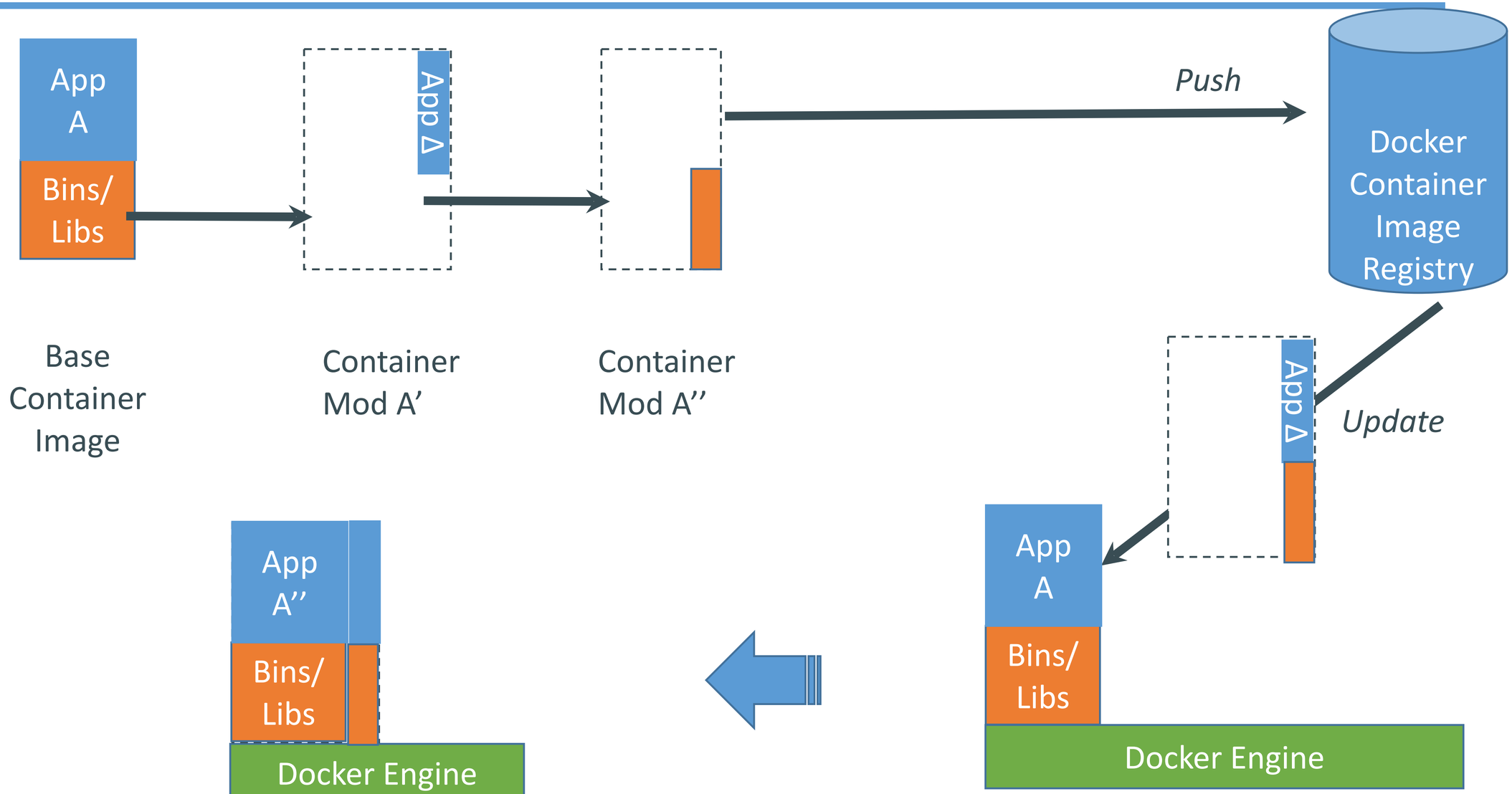
Modified App

Copy on write capabilities allow us to only save the diffs Between container A and container A'

What are the basics of the Docker system?



Changes and Updates



Ecosystem Support

- Operating systems
  - Virtually any distribution with a 2.6.32+ kernel
  - Red Hat/Docker collaboration to make work across RHEL 6.4+, Fedora, and other members of the family (2.6.32 +)
  - CoreOS—Small core OS purpose built with Docker
- OpenStack
  - Docker integration into NOVA (& compatibility with Glance, Horizon, etc.) accepted for Havana release
- Private PaaS
  - OpenShift
  - Solum (Rackspace, OpenStack)
  - Other TBA
- Public PaaS
  - Deis, Voxoz, Cocaine (Yandex), Baidu PaaS
- Public IaaS
  - Native support in Rackspace, Digital Ocean,+++
  - AMI (or equivalent) available for AWS & other
- DevOps Tools
  - Integrations with Chef, Puppet, Jenkins, Travis, Salt, Ansible +++
- Orchestration tools
  - Mesos, Heat, ++
  - Shipyard & others purpose built for Docker
- Applications
  - 1000's of Dockerized applications available at index.docker.io

Use Cases—From Docker Community

Use Case	Examples	Link
Clusters	Building a MongoDB cluster using docker	http://bit.ly/1acbjZf
	Production Quality MongoDB Setup with Docker	http://bit.ly/15CaiHb
	Wildfly cluster using Docker on Fedora	http://bit.ly/1bCIX0O
Build your own PaaS	OpenSource PaaS built on Docker, Chef, and Heroku Buildpacks	http://deis.io
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	http://bit.ly/11BCvvu
	How to put your development environment on docker	http://bit.ly/1b4XtJ3
	Running Drupal on Docker	http://bit.ly/15MJS6B
	Installing Redis on Docker	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	Multi-cloud Deployment with Docker	http://bit.ly/1bF3CN6
Continuous Integration and Deployment	Next Generation Continuous Integration & Deployment with dotCloud's Docker and Strider	http://bit.ly/ZwTfoy
	Testing Salt States Rapidly With Docker	http://bit.ly/1eFBtcm
Lightweight Desktop Virtualization	Docker Desktop: Your Desktop Over SSH Running Inside Of A Docker Container	http://bit.ly/14RYL6x

docker

Want to learn more?

- www.docker.io:
  - Documentation
  - Getting started: interactive tutorial, installation instructions, getting started guide,
  - About: Introductory whitepaper: http://www.docker.io/the-whole-story/
- Github: dotcloud/docker
- IRC: freenode/#docker
- Google groups: groups.google.com/forum/#!forum/docker-user
- Twitter: follow @docker
- Meetups: Scheduled for Boston, San Francisco, Austin, London, Paris, Boulder…and Nairobi. https://www.docker.io/meetups/

docker

docker

www.docker.io