

# Introduction to Docker

November, 2013

---

# Contents

---

- Introduction to Docker, Containers, and the Matrix from Hell
- Why people care: Separation of Concerns
- Technical Discussion
- Ecosystem
- Use Cases
- Docker Futures
- Advanced topics: Networking, Data
- OpenStack
- Learn More



# In the 8 months since we launched

- >200,000 pulls
- >7,500 github stars
- >200 significant contributors
- >200 projects built on top of docker
  - UIs, mini-PaaS, Remote Desktop....
- 1000's of Dockerized applications
  - Memcached, Redis, Node.js...and Hadoop
- Integration in Jenkins, Travis, Chef, Puppet, Vagrant and OpenStack
- Meetups arranged around the world...with organizations like Ebay, Cloudflare, Yandex, and Rackspace presenting on their use of Docker



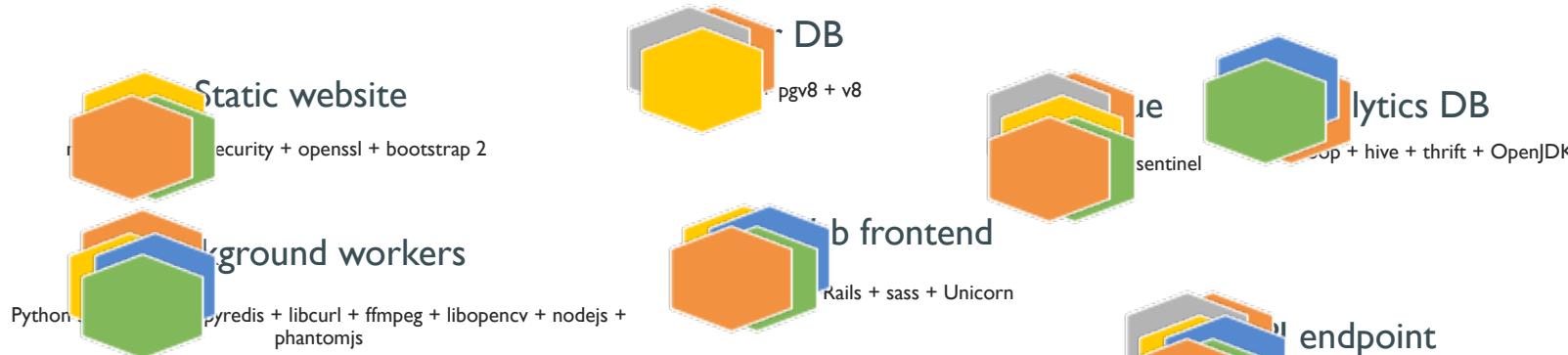
|  |  |
|--|--|
|  <b>David Rousselie</b> @drousselie 2d<br>Docker community is expending. Really the most exciting project lately.<br><a href="http://blog.docker.io/2013/07/docker...">blog.docker.io/2013/07/docker...</a><br><a href="#">Details</a>                      |  <b>Ben Bleything</b> @bleything 5d<br>you guys, @getdocker. holy shit.<br><a href="#">Details</a>  |
|  <b>Phil Whelan</b> @philwhln 2d<br>"Awesome projects from the Docker community   Docker Blog"<br><a href="http://bit.ly/16yC72C">bit.ly/16yC72C</a><br><a href="#">Details</a>   |  <b>omo</b> @omo2009 6d<br><a href="http://blog.docker.io/2013/07/docker...">blog.docker.io/2013/07/docker...</a><br>Docker のなかで X を動かす話。コンテナ作ってから apt-get とか無茶しゃがって....。   |
|  <b>Jake Dahn</b> @jakedahn 6d<br>every time i use @getdocker it just gets more mind-glowingly amazing<br><a href="#">Details</a>   |  <b>Sandeep</b> @machbio 23d<br>One of the most Kick-ass Project at this Moment.. credits to @programm and #docker.io<br><a href="#">Details</a>  |
|  <b>Luc Perkins</b> @lucperkins 2d<br>Somehow I get this weird feeling that I haven't even begun to grasp the implications of @getdocker<br><a href="#">Details</a> <a href="#">Reply</a> <a href="#">Retweet</a> <a href="#">Star</a> <a href="#">More</a> |  <b>John Fink</b> @adr 3d<br>there are probably a million of these, but this one is mine: generic LAMP stack for @getdocker.<br><a href="http://index.docker.io/u/jbfink/lamps...">index.docker.io/u/jbfink/lamps...</a><br><a href="#">Details</a> <a href="#">Reply</a> <a href="#">Retweet</a> <a href="#">Star</a> <a href="#">More</a> |
|  <b>Phil Plante</b> @pplante 23d<br>woot! our new @getdocker cluster is performing way better than expected, and is 5x faster than our cloud setup.<br><a href="#">Details</a>  |  <b>Damian Gryski</b> @dgryski 3d<br>. @i_x_s All the cool kids are moving towards @getdocker.<br><a href="#">Conversation</a>  |
|  |  <b>Fenn</b> @fennb 24d<br>Docker (& LXC in general) could be the most important step in virtualization since hypervisors. Impressive stuff: <a href="http://docker.io">docker.io</a><br><a href="#">Details</a>  |

Why all the excitement?

---

# The Challenge

M  
u  
l  
t  
i  
p  
l  
i  
c  
i  
v  
t

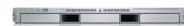


Do services and apps interact appropriately?

Multiplicity of hardware environments



Development VM

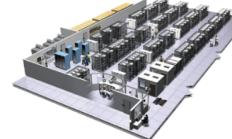


QA server

Customer Data Center



Public Cloud



Production Cluster



Disaster recovery

Contributor's laptop



Production Servers

Can I migrate smoothly and quickly ?

# The Matrix From Hell

| 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  |
|                    |  |  |  |  |  |  |  |



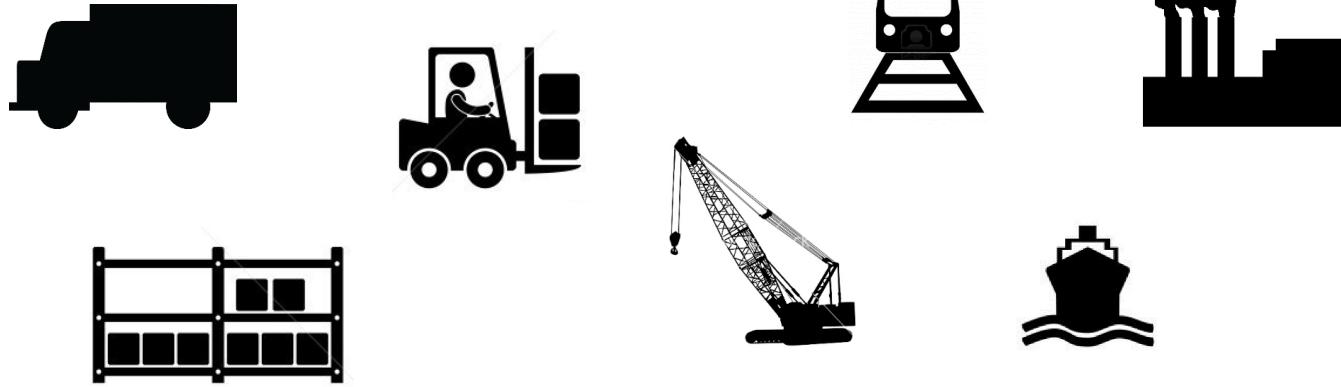
# Cargo Transport Pre-1960

M  
u  
lt  
i  
p  
l  
i  
c  
i  
t  
v



Do I  
worry  
about  
how  
goods  
interact  
(e.g.  
coffee)

Multiplicity  
of  
methods  
for  
transporting/  
storing



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



# Also a matrix from hell

---

|  |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|
|  | ? | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? | ? |
|  |   |   |   |   |   |   |   |

# Solution: Intermodal Shipping Container

M  
u  
l  
t  
i  
p  
l  
i  
c  
i  
t  
y  
o  
f  
G  
o

Multiplicity of methods for transporting/storing



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

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

# Docker is a shipping container system for code

M  
u  
l  
t  
i  
p  
l  
i  
c  
i  
t  
y  
o  
f  
S  
ta



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

Do services and apps interact appropriately?

M  
u  
l  
t  
i  
p  
l  
i  
c  
i  
t  
y  
o  
f  
h  
a  
r  
d  
w  
a  
r  
e  
e  
n  
v  
i  
r  
o  
n  
m  
e  
n  
t  
s



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

Can I migrate smoothly and quickly?



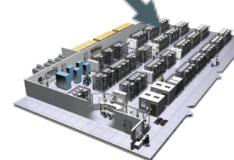
Development VM



QA server



Customer Data Center



Public Cloud



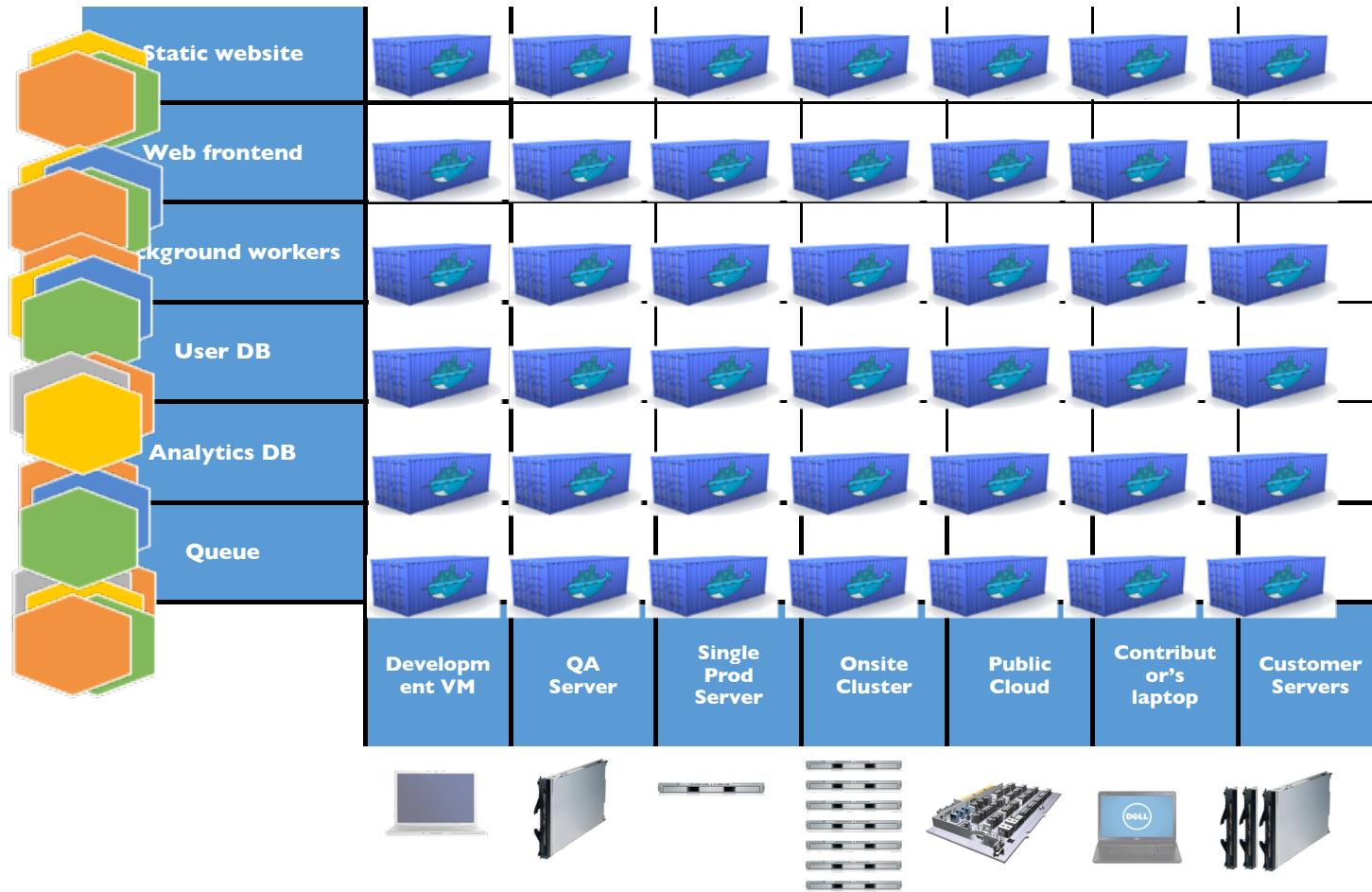
Production Cluster



Contributor's laptop



# 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)



# Why Devops Cares?

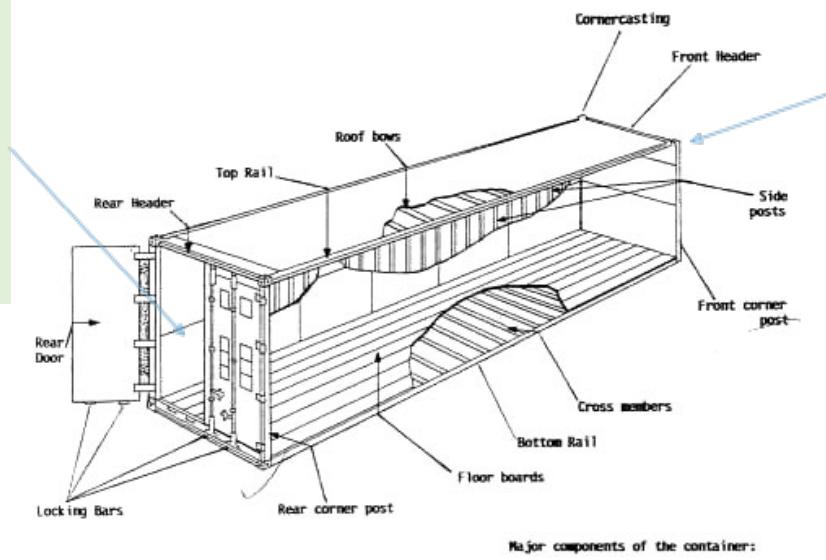
---

- 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



# 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



- 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

---

## WH

### Y

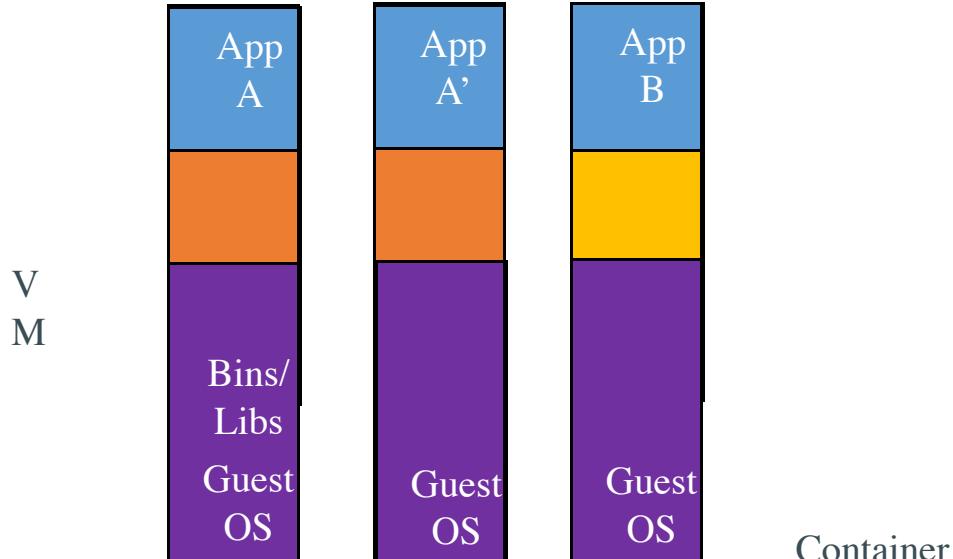
- Run everywhere
  - Regardless of kernel version (2.6.32+)
  - Regardless of host distro
  - Physical or virtual, cloud or not
  - Container and host architecture must match\*
- Run anything
  - If it can run on the host, it can run in the container
  - i.e. if it can run on a Linux kernel, it can run

## WHA

- 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>>

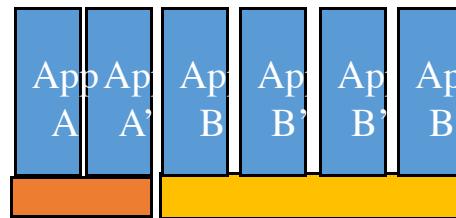


# Containers vs. VMs

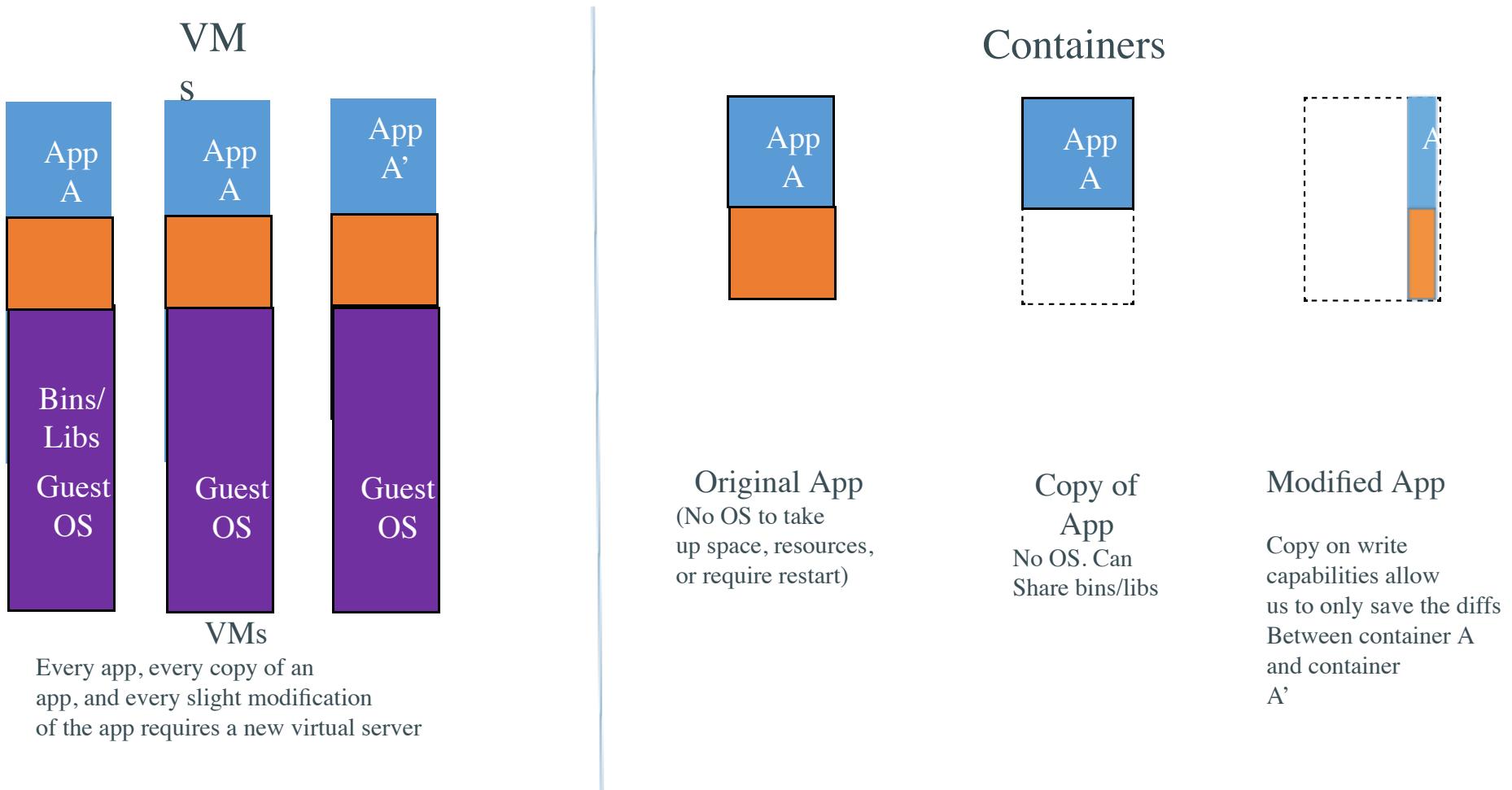


Containers are isolated, but share OS and, where appropriate, bins/libraries

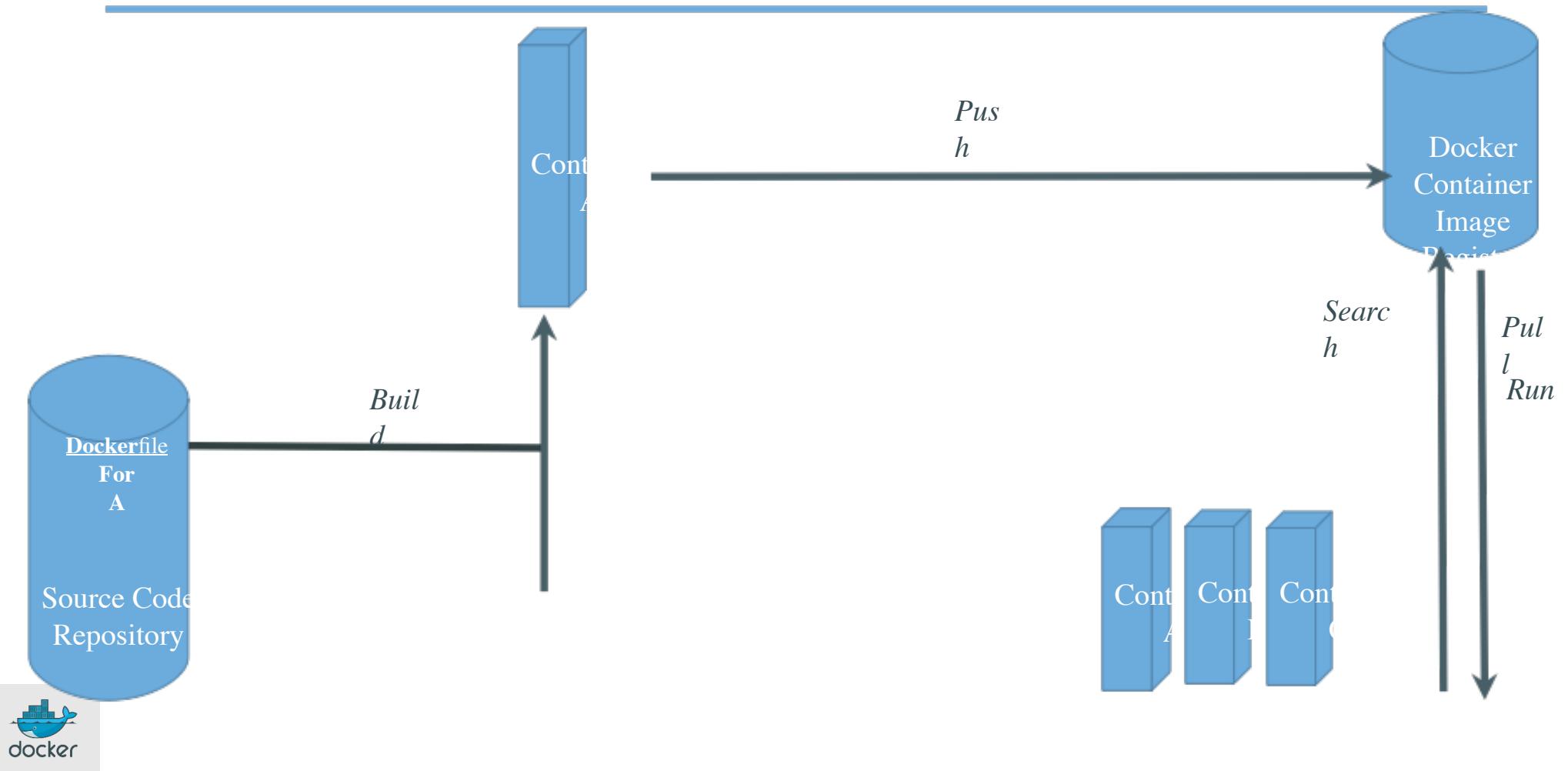
...result is significantly faster deployment, much less overhead, easier migration, faster restart



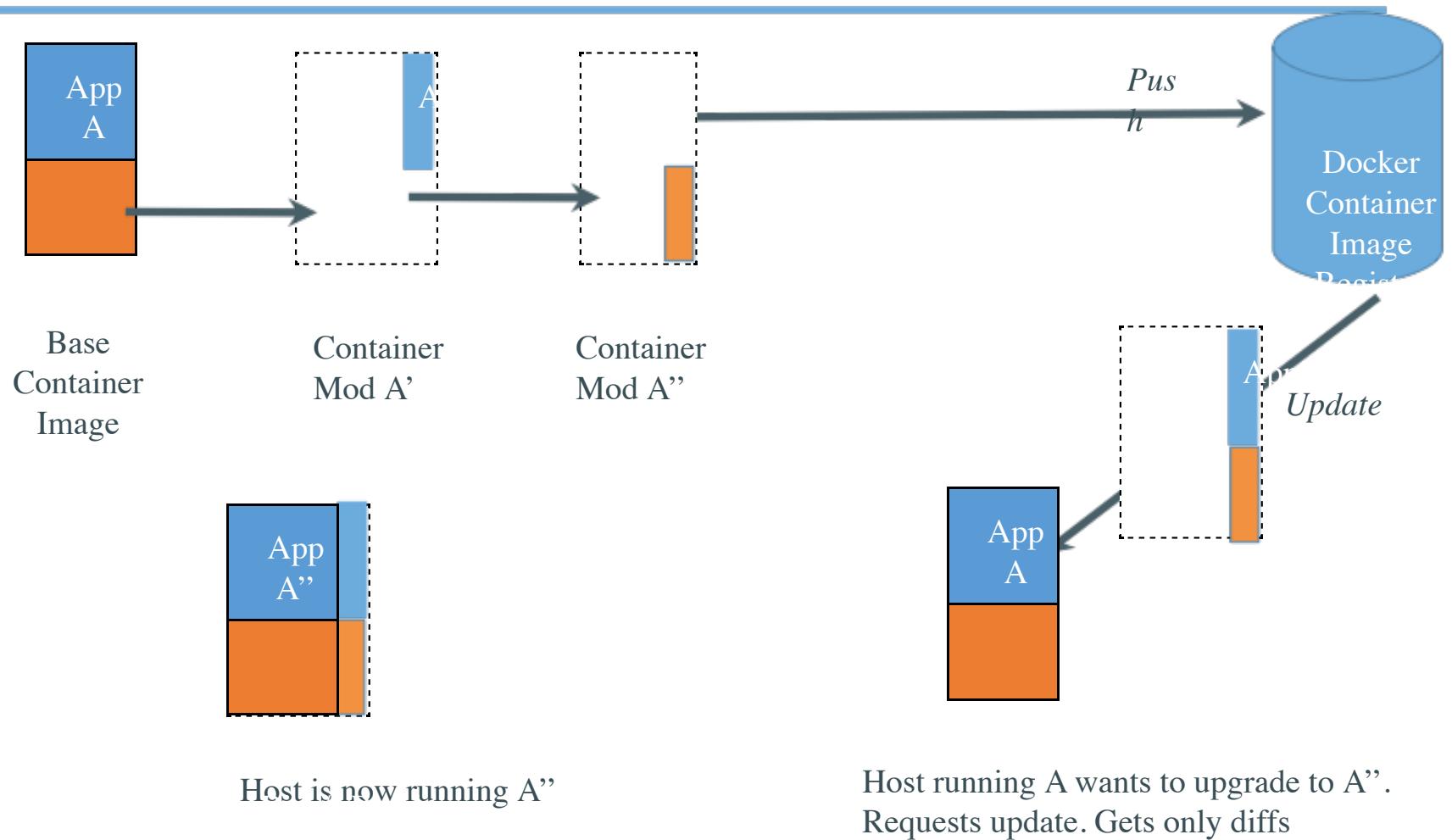
# Why are Docker containers lightweight?



# 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 R other members of the family (2.6.32 +)
    - CoreOS—Small core OS purpose built with Docker
  - OpenStack
    - Docker integration into NOVA (& `openstack`ility with Glance, Horizon, etc.) accepted for Havana release
  - Private PaaS
    - OpenShift
    - Solum (Rackspace, OpenStack)
    - Other TBA
  - Public PaaS
    - Deis, Voxoz, Cocaine (Yandex), Baidu PaaS
- 
- The slide features several logos of ecosystem partners arranged in two columns. The first column includes DigitalOcean (blue cloud with dots), Rackspace (red square with white 'r'), and OpenShift (red circle with white 'o'). The second column includes Chef (orange 'c' with red dot), Jenkins (Jenkins logo with a cartoon character), Red Hat (red hat icon), and Yandex (red and white stylized 'y' logo).



# Use Cases

---

- Ted Dziuba on the Use of Docker for Continuous Integration at Ebay Now
  - <https://speakerdeck.com/teddziuba/docker-at-ebay>
  - [http://www.youtube.com/watch?feature=player\\_embedded&v=0Hi0W4gX--4](http://www.youtube.com/watch?feature=player_embedded&v=0Hi0W4gX--4)
- Sasha Klizhentas on use of Docker at Mailgun/Rackspace
  - [http://www.youtube.com/watch?feature=player\\_embedded&v=CMC3xdAo9RI](http://www.youtube.com/watch?feature=player_embedded&v=CMC3xdAo9RI)
- Sebastien Pahl on use of Docker at CloudFlare
  - [http://www.youtube.com/watch?feature=player\\_embedded&v=-Lj3jt\\_-3r0](http://www.youtube.com/watch?feature=player_embedded&v=-Lj3jt_-3r0)
- Cambridge HealthCare
  - <http://blog.howareyou.com/post/62157486858/continuous-delivery-with-docker-and-jenkins-part-i>
- Red Hat Openshift and Docker
  - <https://www.openshift.com/blogs/technical-thoughts-on-openshift-and-docker>

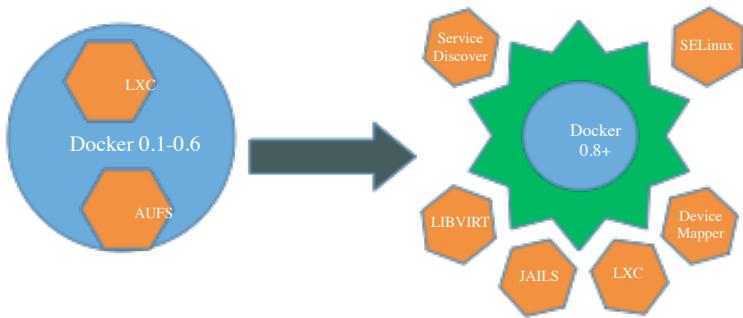


# Use Cases—From Our Community

| Use Case                              | Examples   | Link  |
|---------------------------------------|--|---|
| Clusters                              | Building a MongoDB cluster using docker  | <a href="http://bit.ly/1acbJZf">http://bit.ly/1acbJZf</a> |
|                                       | Production Quality MongoDB Setup with Docker   | <a href="http://bit.ly/15CaiHb">http://bit.ly/15CaiHb</a> |
|                                       | Wildfly cluster using Docker on Fedora   | <a href="http://bit.ly/1bClX0O">http://bit.ly/1bClX0O</a> |
| Build your own PaaS                   | OpenSource PaaS built on Docker, Chef, and Heroku Buildpacks                                       | <a href="http://deis.io">http://deis.io</a>               |
| Web Based Environment for Instruction | JiffyLab – web based environment for the instruction, or lightweight use of, Python and UNIX shell | <a href="http://bit.ly/12oaj2K">http://bit.ly/12oaj2K</a> |
| Easy Application Deployment           | Deploy Java Apps With Docker = Awesome   | <a href="http://bit.ly/11BCvvu">http://bit.ly/11BCvvu</a> |
|                                       | How to put your development environment on docker  | <a href="http://bit.ly/1b4XtJ3">http://bit.ly/1b4XtJ3</a> |
|                                       | Running Drupal on Docker   | <a href="http://bit.ly/15MJS6B">http://bit.ly/15MJS6B</a> |
|                                       | Installing Redis on Docker   | <a href="http://bit.ly/16EWOKh">http://bit.ly/16EWOKh</a> |
| Create Secure Sandboxes               | Docker makes creating secure sandboxes easier than ever  | <a href="http://bit.ly/13mZGJH">http://bit.ly/13mZGJH</a> |
| Create your own SaaS                  | Memcached as a Service   | <a href="http://bit.ly/11nL8vh">http://bit.ly/11nL8vh</a> |
| Automated Application Deployment      | Multi-cloud Deployment with Docker   | <a href="http://bit.ly/1bF3CN6">http://bit.ly/1bF3CN6</a> |
| Continuous Integration and Deployment | Next Generation Continuous Integration & Deployment with dotCloud's Docker and Strider             | <a href="http://bit.ly/ZwTfoy">http://bit.ly/ZwTfoy</a>   |
|                                       | Testing Salt States Rapidly With Docker  | <a href="http://bit.ly/1eFBtcm">http://bit.ly/1eFBtcm</a> |
| Lightweight Desktop                   | <a href="#">Docker Desktop: Your Desktop Over SSH Running Inside Of A Docker Container</a>         | <a href="http://bit.ly/14RYL6x">http://bit.ly/14RYL6x</a> |



# Docker Futures\*



- Docker 0.7 (current release)
  - Fedora compatibility
  - Reduce kernel dependencies
  - Device mapper
  - Container linking
- Docker 0.8 (Dec)
  - Shrink and stabilize Core
  - Provide stable, pluggable API
  - RHEL compatibility
  - Nested containers
  - Beam: Introspection API based on Redis
  - expand snapshot management features for data volumes
  - We will consider this “production ready”
- Docker 0.9 (Jan)
- Docker 1.0 (Feb)
  - We will offer support for this product

\* We shoot for time based releases (1x/5wks), features are targeted, but not guaranteed for particular releases



# Advanced topics

---

- Data

- Today: Externally mounted volumes
  - Share volumes between containers
  - Share volume between a containers and underlying hosts
    - high-performance storage backend for your production database
    - making live development changes available to a container, etc.
  - Optional: specify memory limit for containers, CPU priority
  - Device mapper/ LVM snapshots in 0.7

- Futures:

- I/O limits
- Container resource monitoring (CPU & memory usage)
- Orchestration (linking & synchronization between containers)
- Cluster orchestration (multi-host environment)

- Networking

- Supported today:
  - UDP/TCP port allocation to containers



---

# OpenStack / Docker

New hypervisor to enable Nova to deploy Linux containers



# Why Docker + OpenStack

---

- Alternative to VMs within OpenStack-today
- Easier deployment of OpenStack itself-near future
- Cross cloud application deployment
- At OpenStack Summit we will show:
  - Building and testing an application from source
  - Running on a laptop
  - Running it, without modification or noticeable downtime, on a public cloud
  - Running it, without modification or noticeable downtime, on an openstack cluster
  - Doing all of the above using Nova, Glance and Horizon
- Containers orchestration with OpenStack Heat (Demo at summit)

