An Introduction to Containers and the Docker Engine

Joshua Bradley | @jbradley888

nebulaworks | @nebulaworks

Container Technology Overview

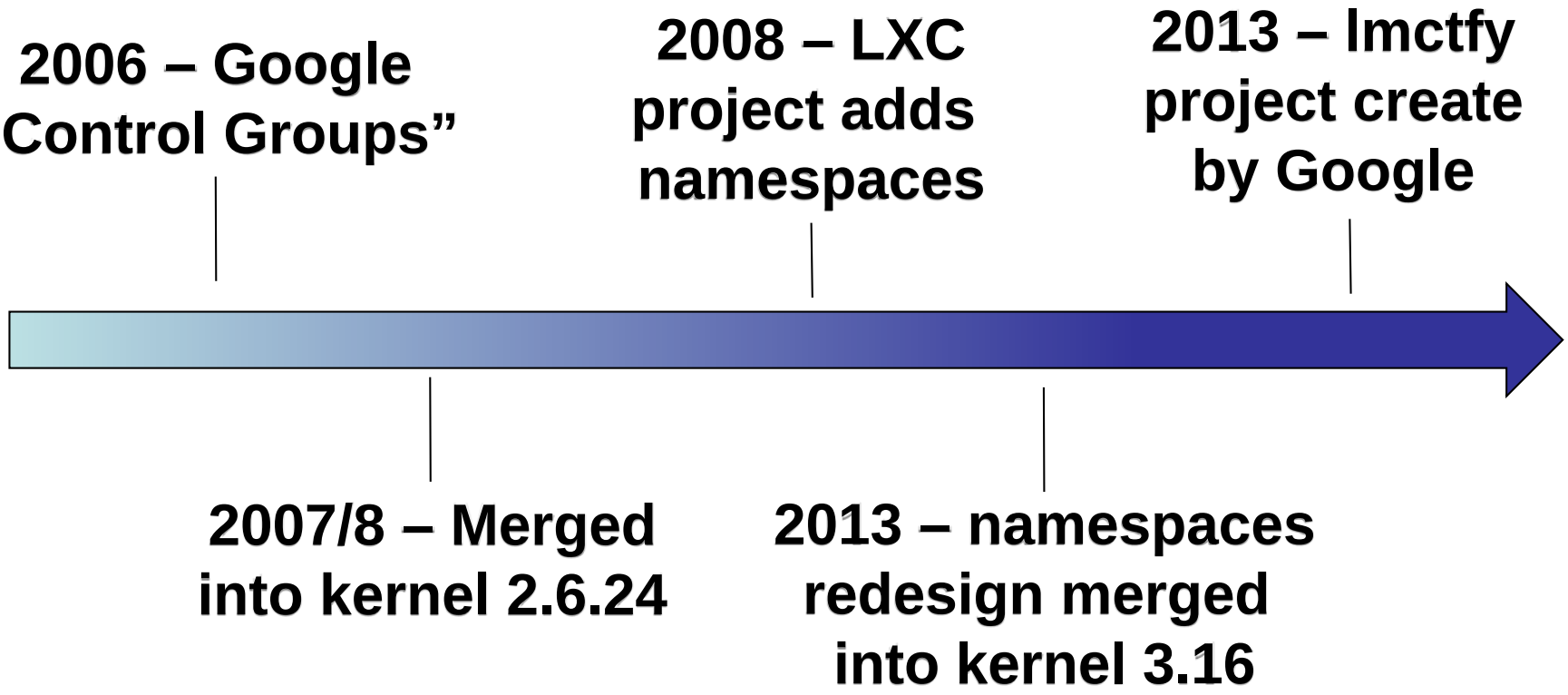
Before the Linux Container

Solaris Zones

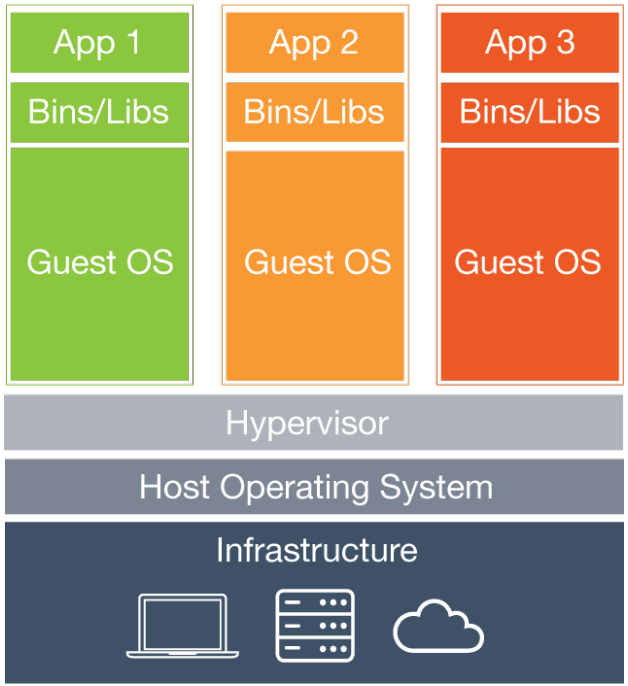
BSD Jails

LPARs

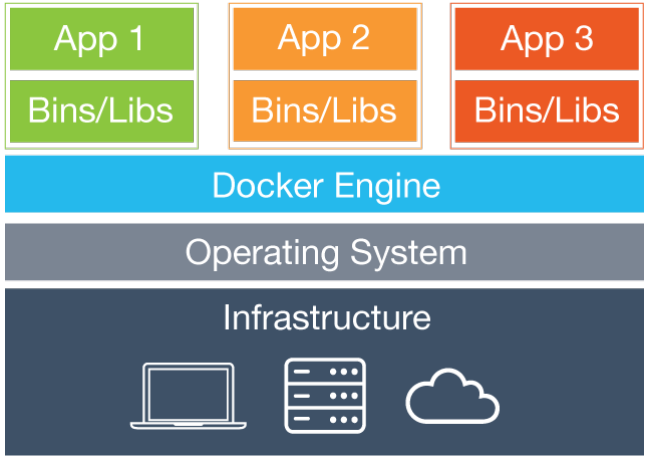
History of the Linux Container



VM vs Containers



VS



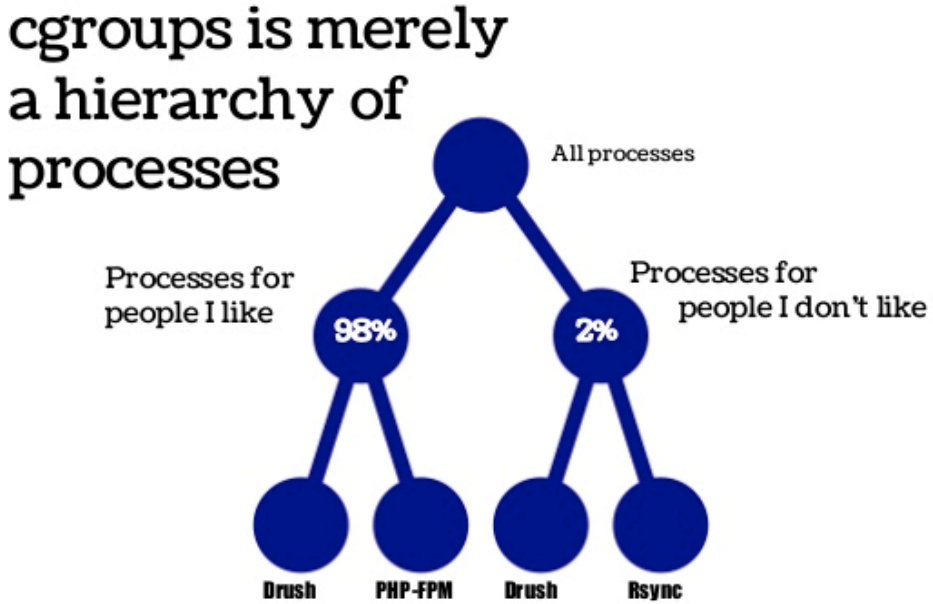
Benefits of Process Virtualization

- Density
- Provisioning
- Performance

Drawbacks of Process Virtualization

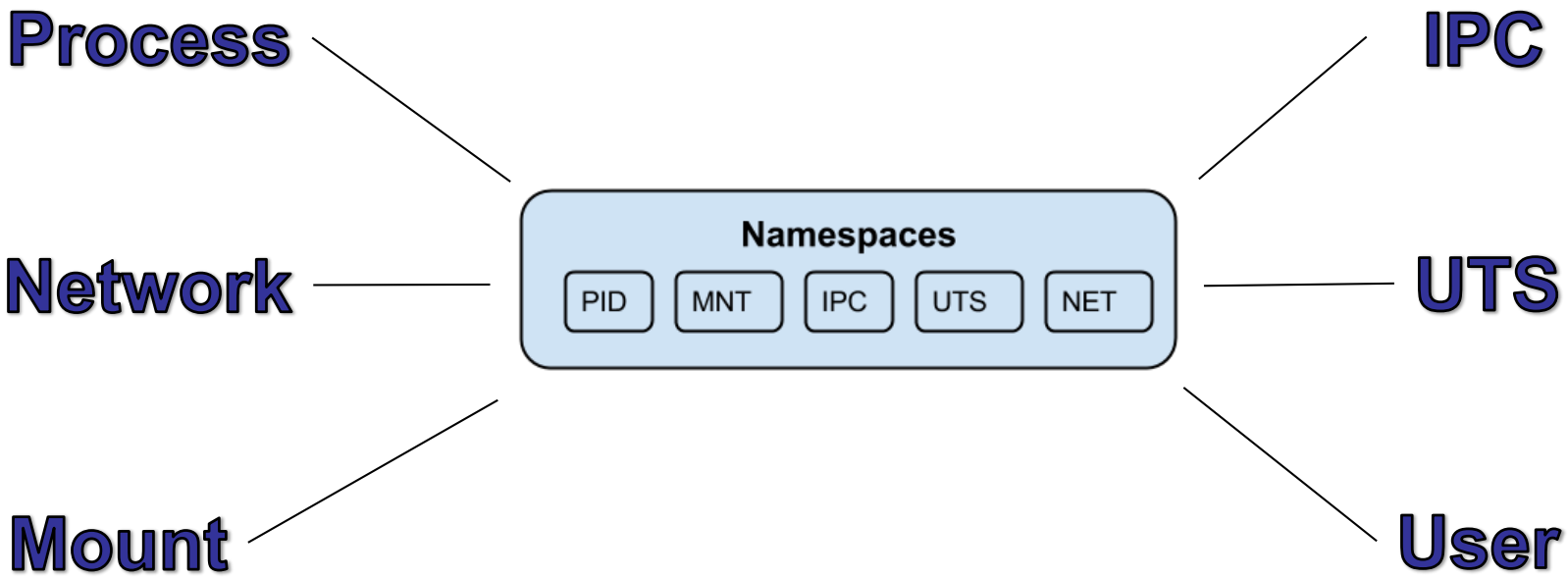
- Reduced isolation
- Reduced security

C-Groups



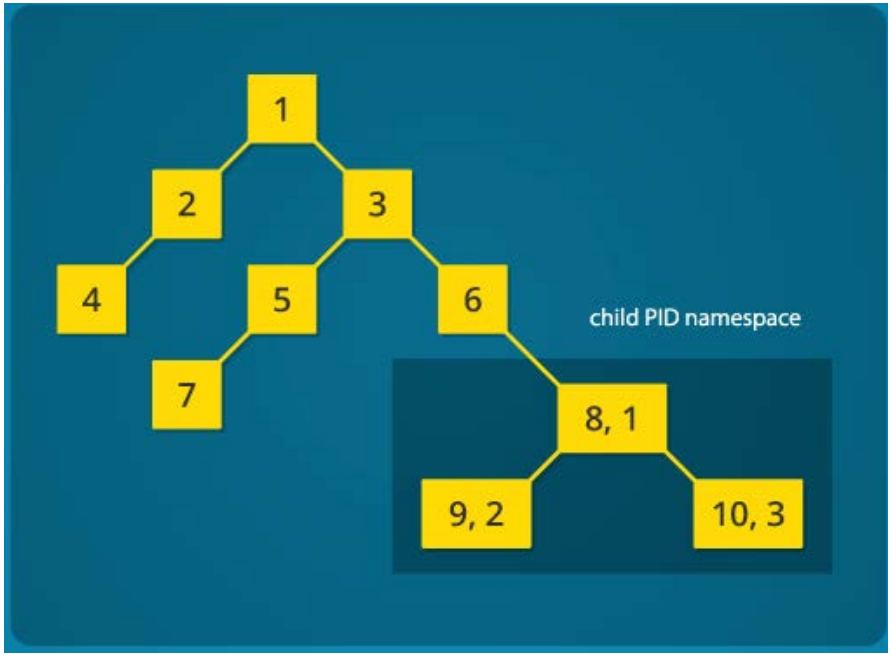
- Resource limitation
- Prioritization
- Accounting
- Control

Namespaces


Process
Network
Mount
IPC
UTS
User
Namespaces
PID
MNT
IPC
UTS
NET

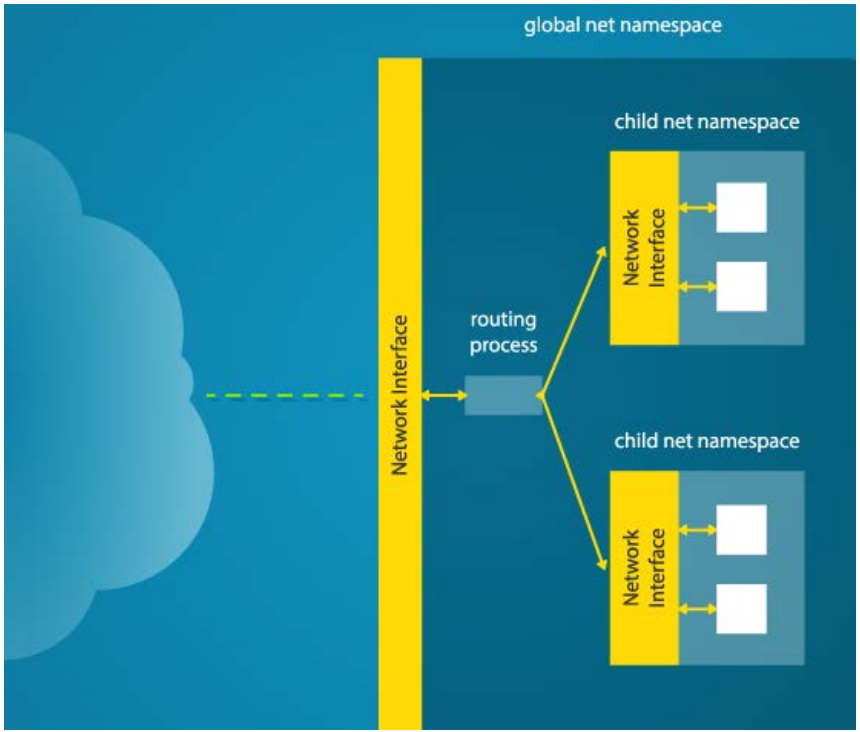
Process Namespace

The Process namespace provides isolation for the allocation of process identifiers (PIDs).



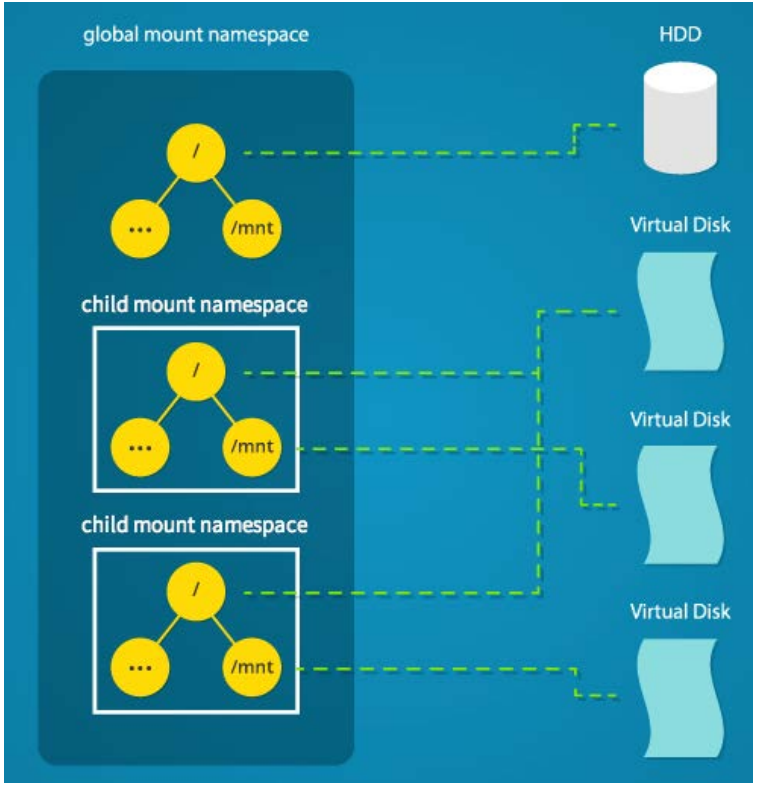
Network Namespace

The Network namespace isolates the network interface controllers (physical or virtual), iptables firewall rules, routing tables etc.

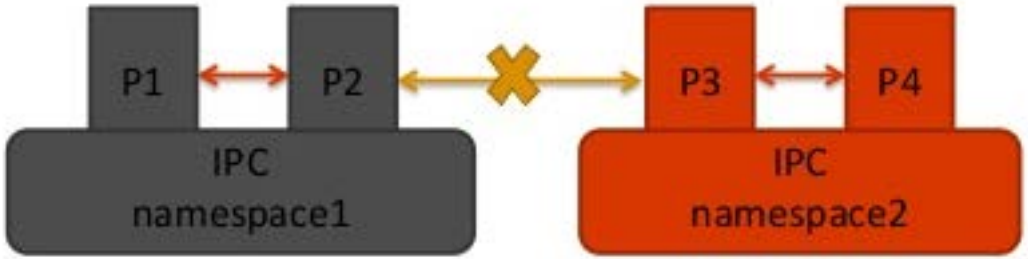


Mount Namespace

The Mount namespace allows the creation of a different file system layout, or making certain mount points read-only.



IPC Namespace

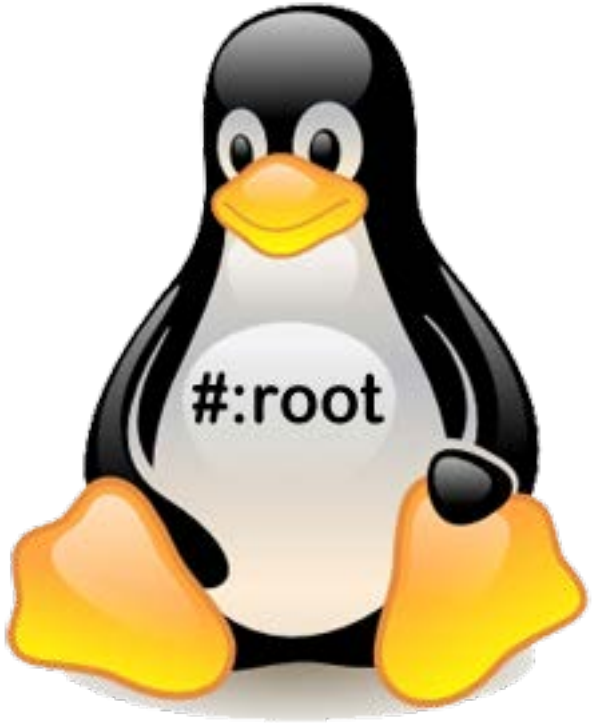


The IPC namespace isolates the System V inter-process communication between namespaces.

UTC Namespace

The "UTS" namespace allows changing the hostname.

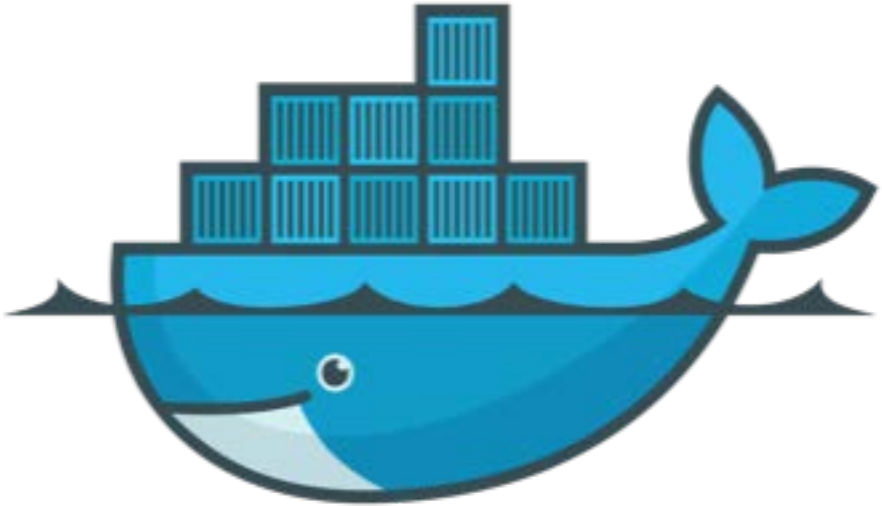
User Namespace



The User namespace isolates the user IDs between namespaces.

Docker Architecture

Docker who?

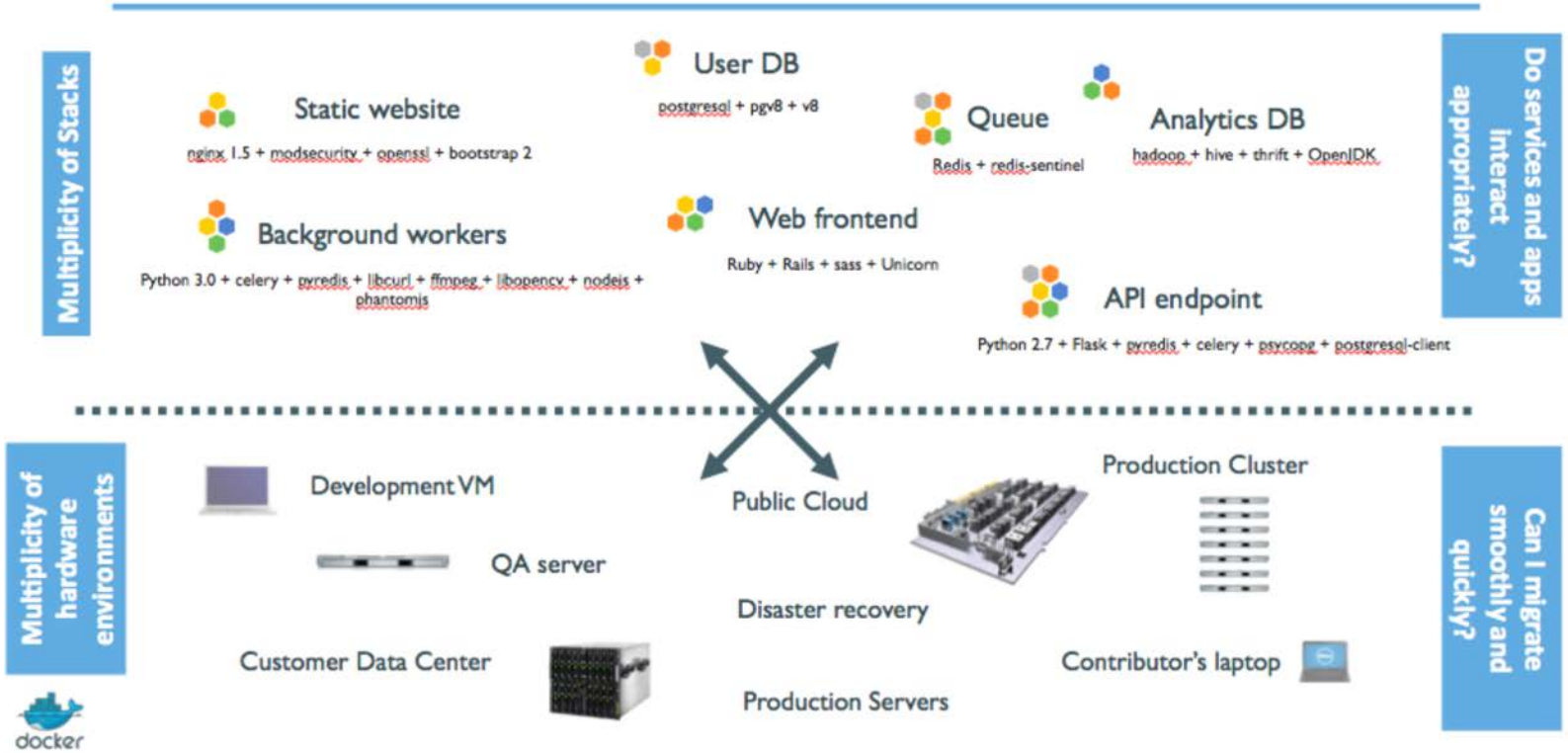


Company | Project | Container

Docker, Inc. a history ….

JAN-2013	dotCloud starts Docker project
AUG-2013	eBay, Uber, Cloudflare start using Docker
OCT-2013	Docker, Inc. splits from dotCloud
DEC-2013	Google CE announces support for Docker
APR-2014	AWS announces support for Docker
JUN-2014	Docker 1.0 released @ dockercon
SEP-2014	$40 Million – C round funding
OCT-2014	Partnership with Microsoft announced
DEC-2014	Docker announces Machine, Compose, Swarm

The Challenge

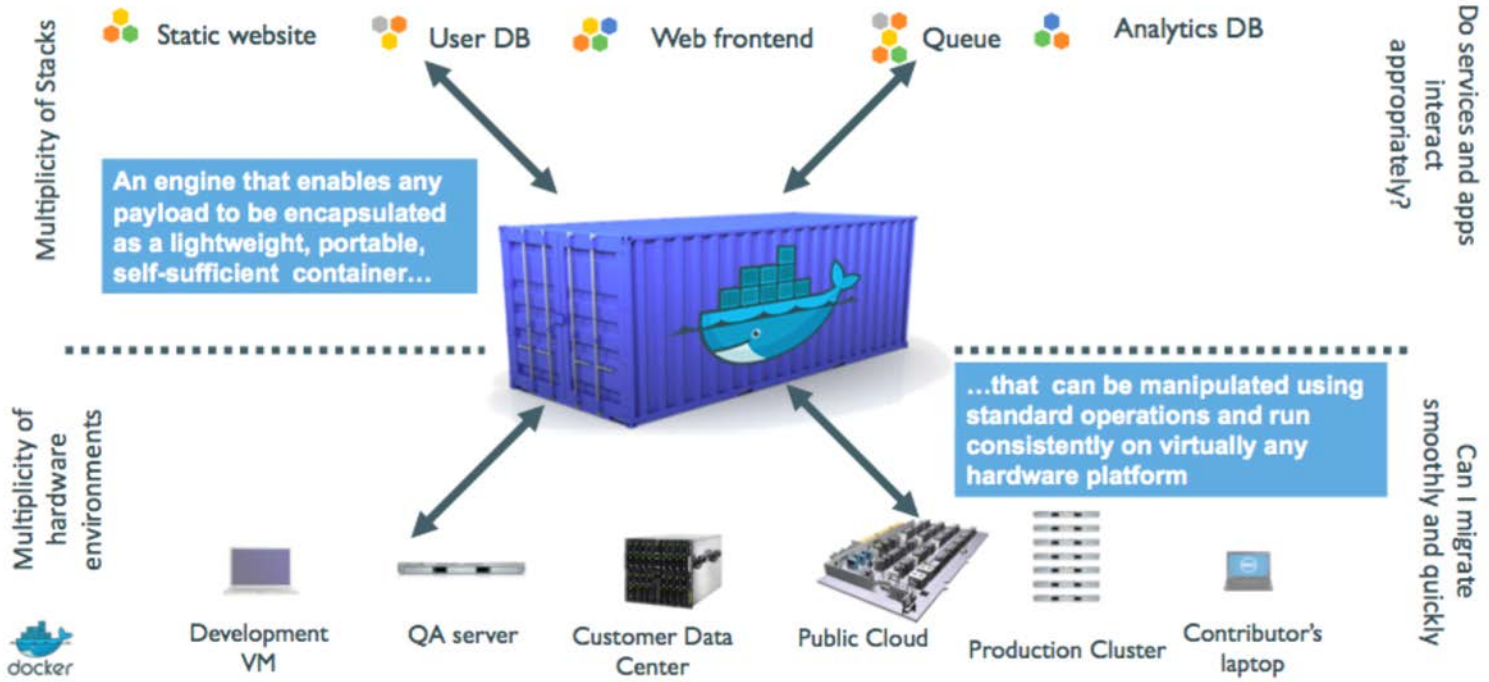


Finding Inspiration

450,000

45,000

Container for Apps

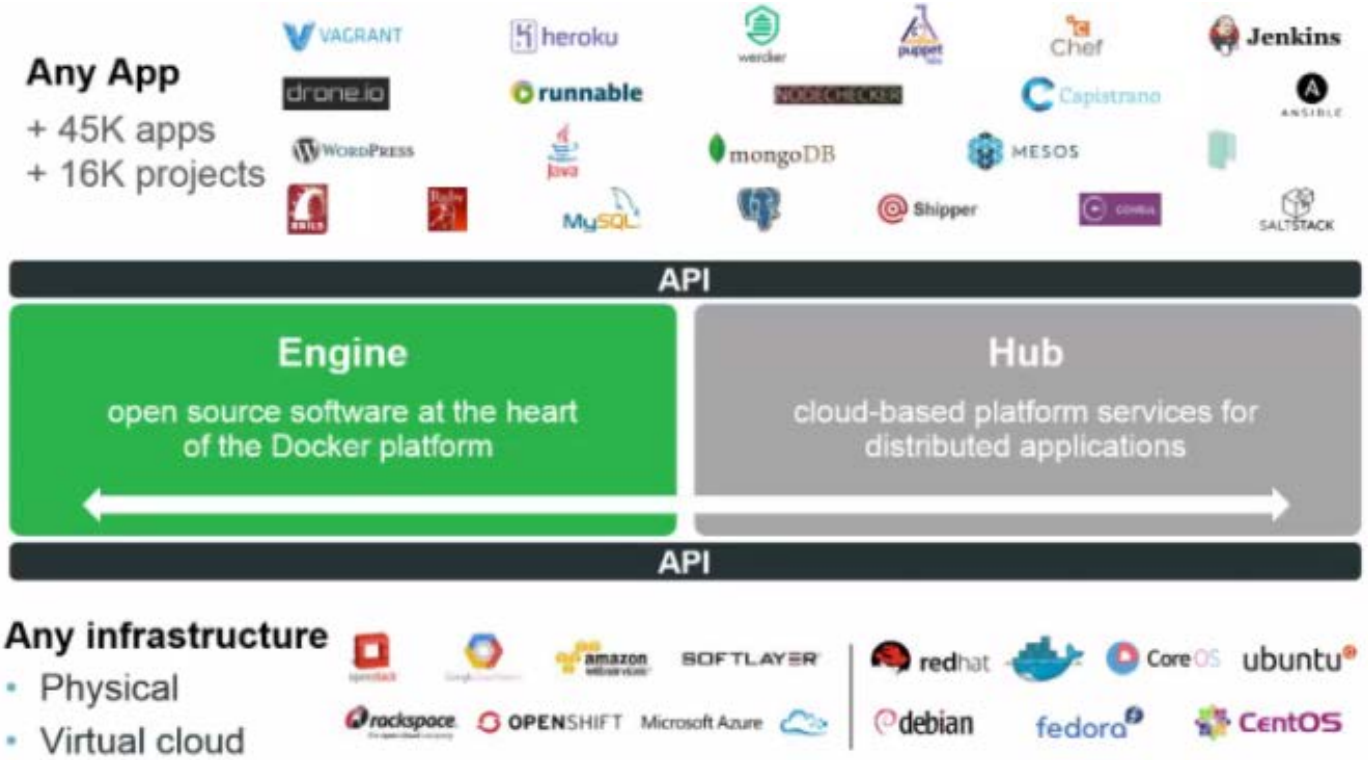


Yeah…but why Docker Containers?

The Project

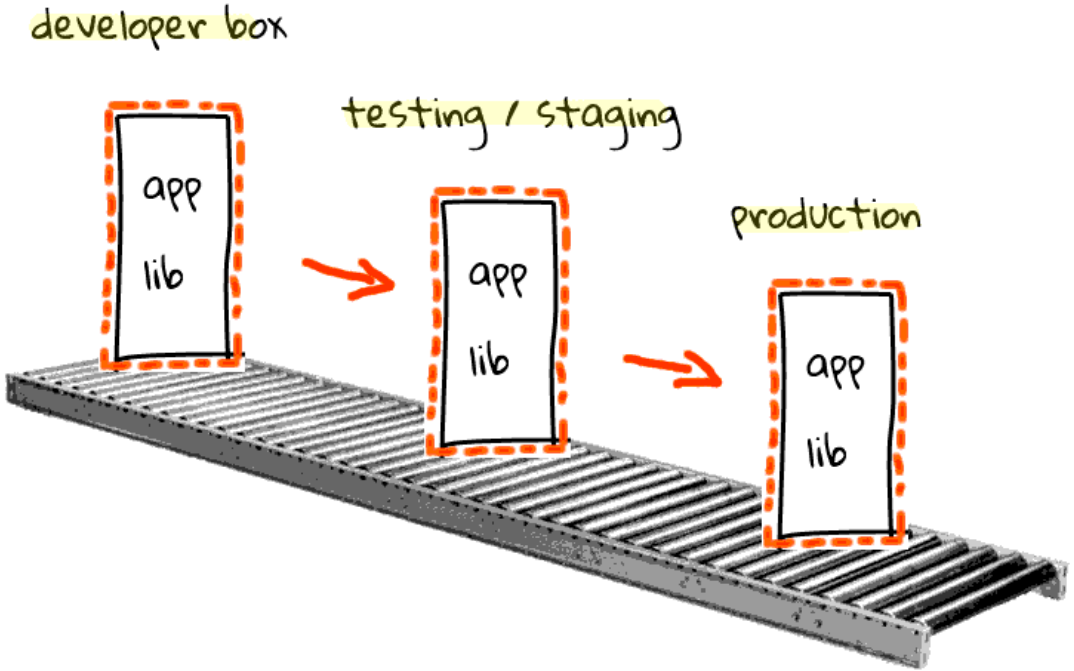
- 1000+ contributors
- 450,000,000 image downloads last year
- 100,000+ hosted Docker projects
- 90+ “official” Joint projects

The Ecosystem

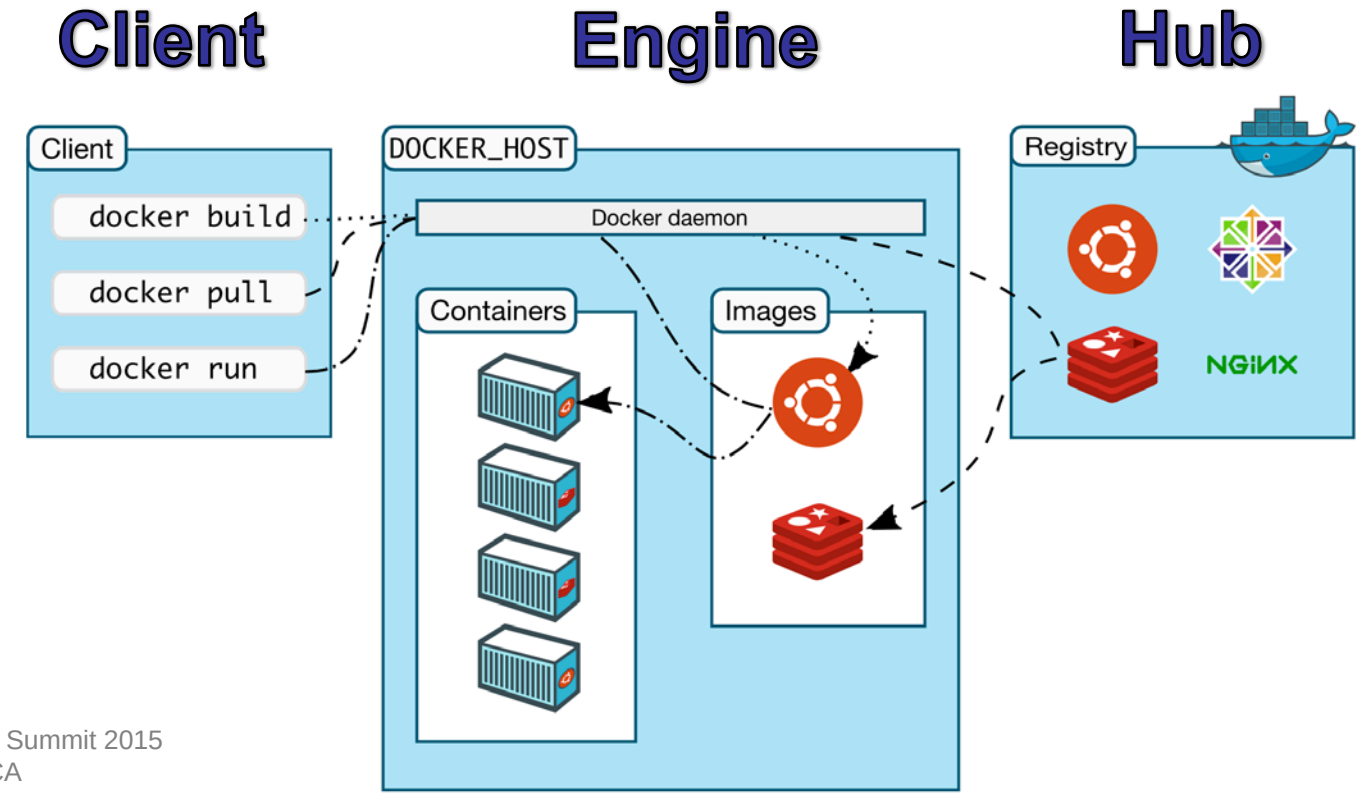


The Expanded Project

Still…why the fuss?

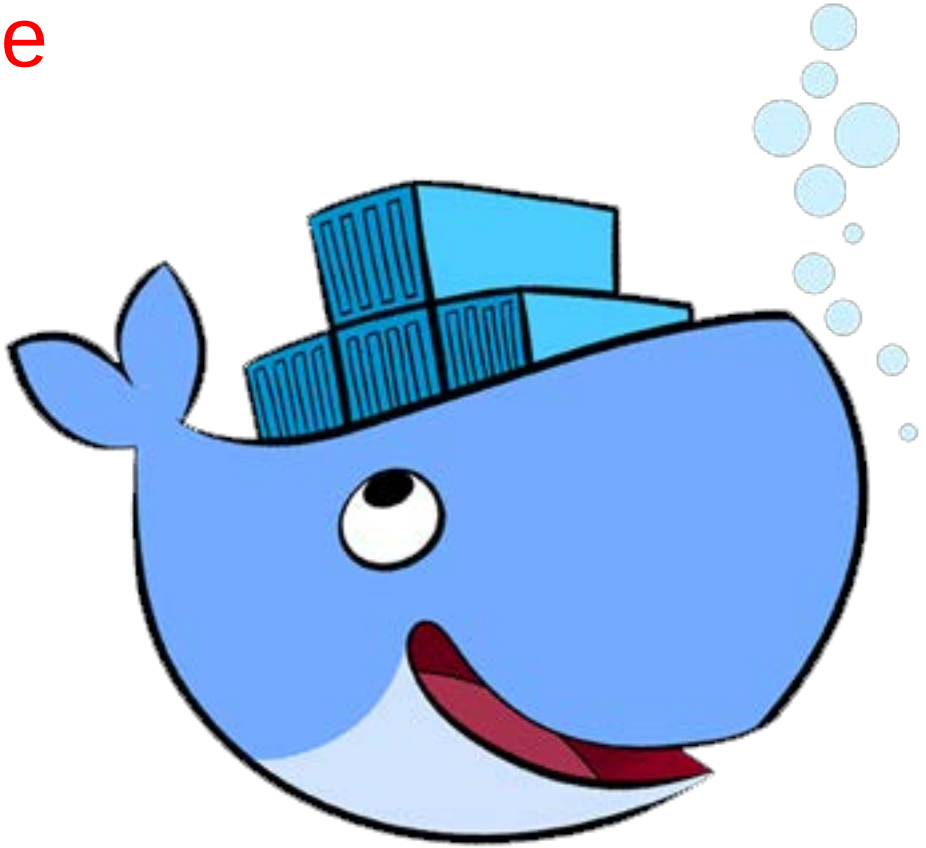


Docker architecture



Docker Engine

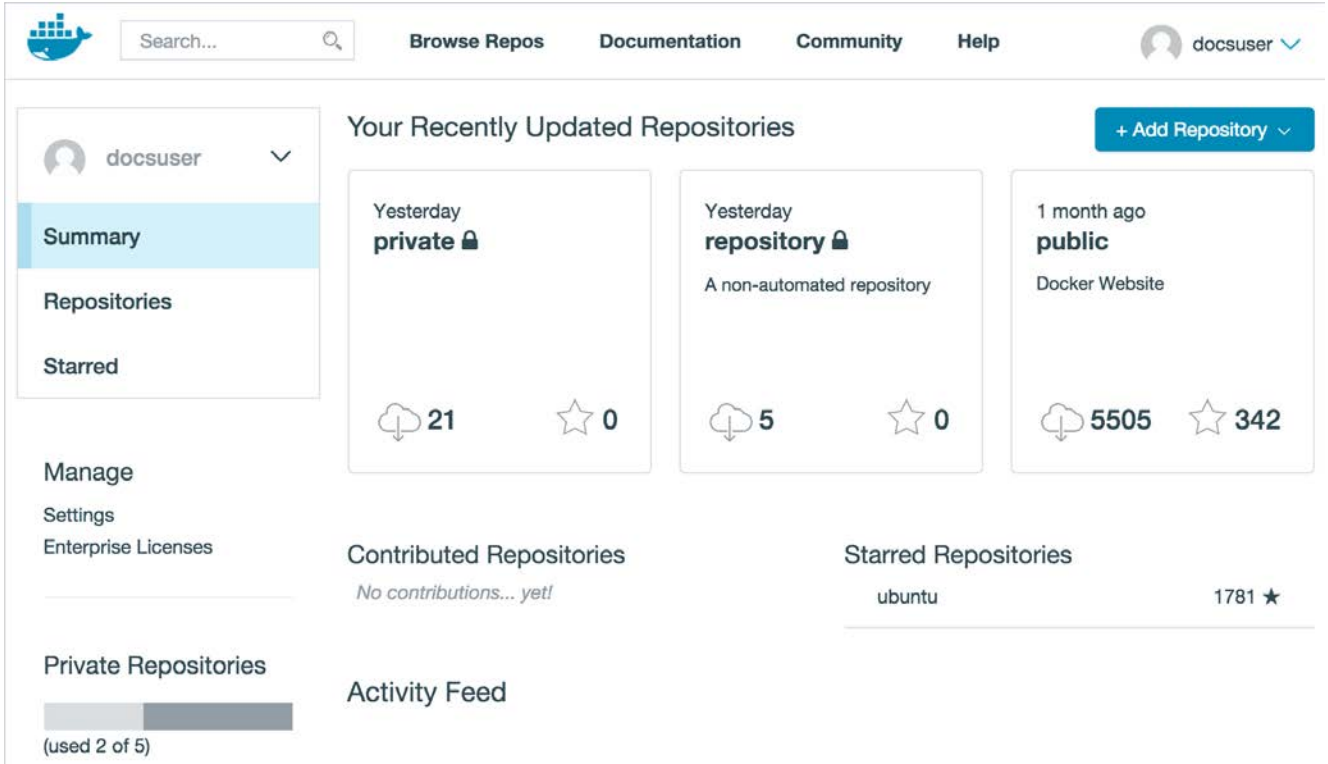
- Create Containers
- Manages Containers
- Build Images
- Manage Local Images



Docker Client

- Cross Platform
  - OSX
  - Windows
  - Linux
- Local/Remote Execution
- Rest API

Docker Hub

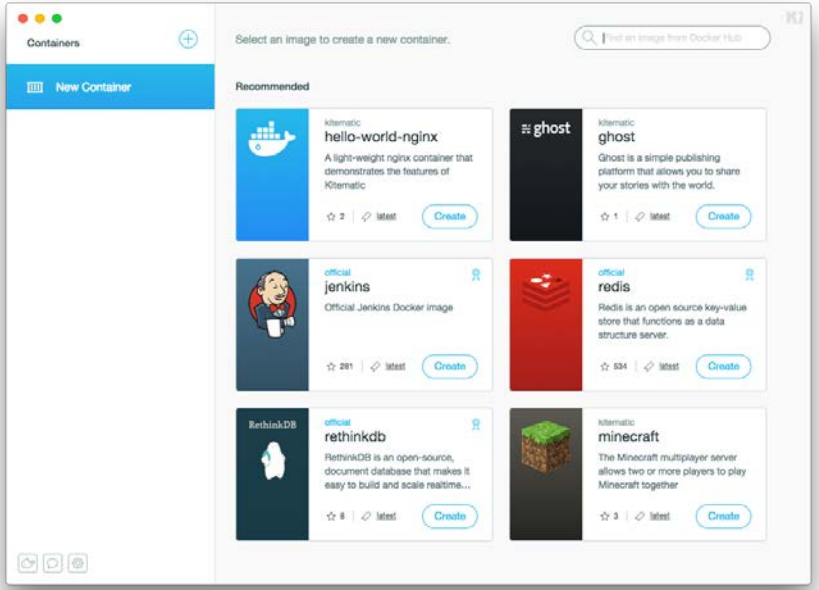

Search...
Browse Repos
Documentation
Community
Help
docsuser
docsuser
Summary
Repositories
Starred
Manage
Settings
Enterprise Licenses
Private Repositories
(used 2 of 5)
Your Recently Updated Repositories
+ Add Repository
Yesterday
private
21
0
Yesterday
repository
A non-automated repository
5
0
1 month ago
public
Docker Website
5505
342
Contributed Repositories
No contributions... yet!
Starred Repositories
ubuntu
1781 ★
Activity Feed

How to get Docker?

- Linux installers
- Script from docker > http://get.docker.com
- Get the binary > https://*get.docker.com/builds/Linux/x86_64/docker-latest*
- Boot2docker or Kitematic

Boot2docker vs Kitematic

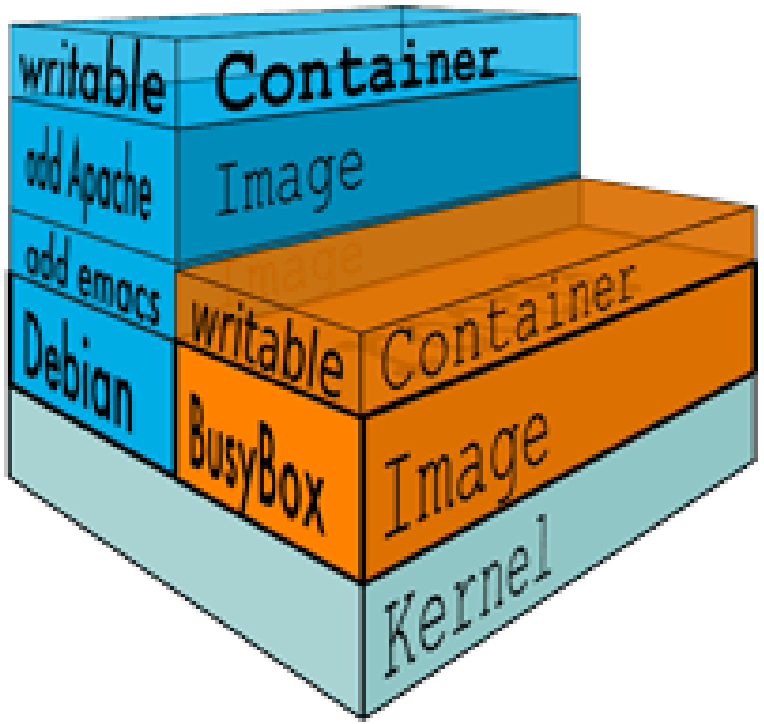
VS



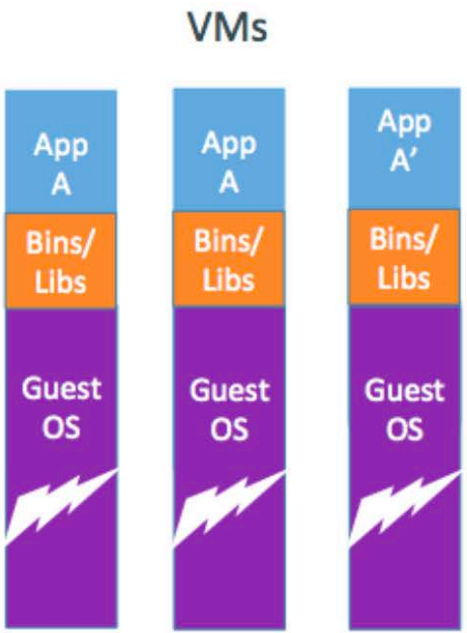
DEMO TIME

Containers enabling CI/CD

Docker Image format

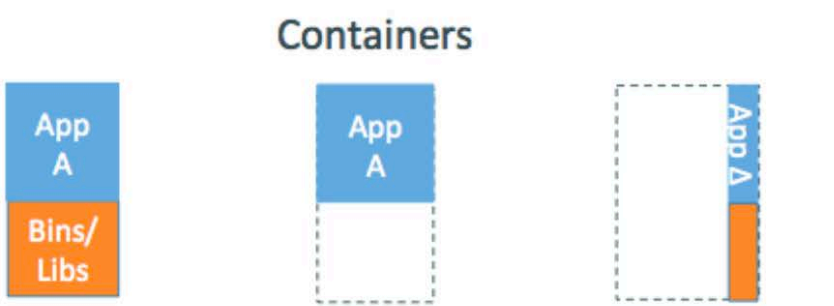


Easy to Ship



VMs

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



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

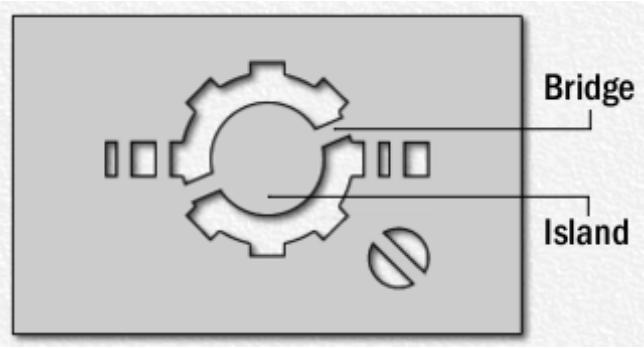
Copy of App
No OS. Can Share bins/libs

Modified App

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

Images from layers

- An image is a collection of files + some meta data
- Images are made of linked layers.
- Each layer can add, change, and remove files.
- Images can share layers to optimize disk usage, transfer times, and memory use.



Building images

- docker commit – manual process
- docker build – automated build
- docker import – importing a filesystem

Docker image namespaces

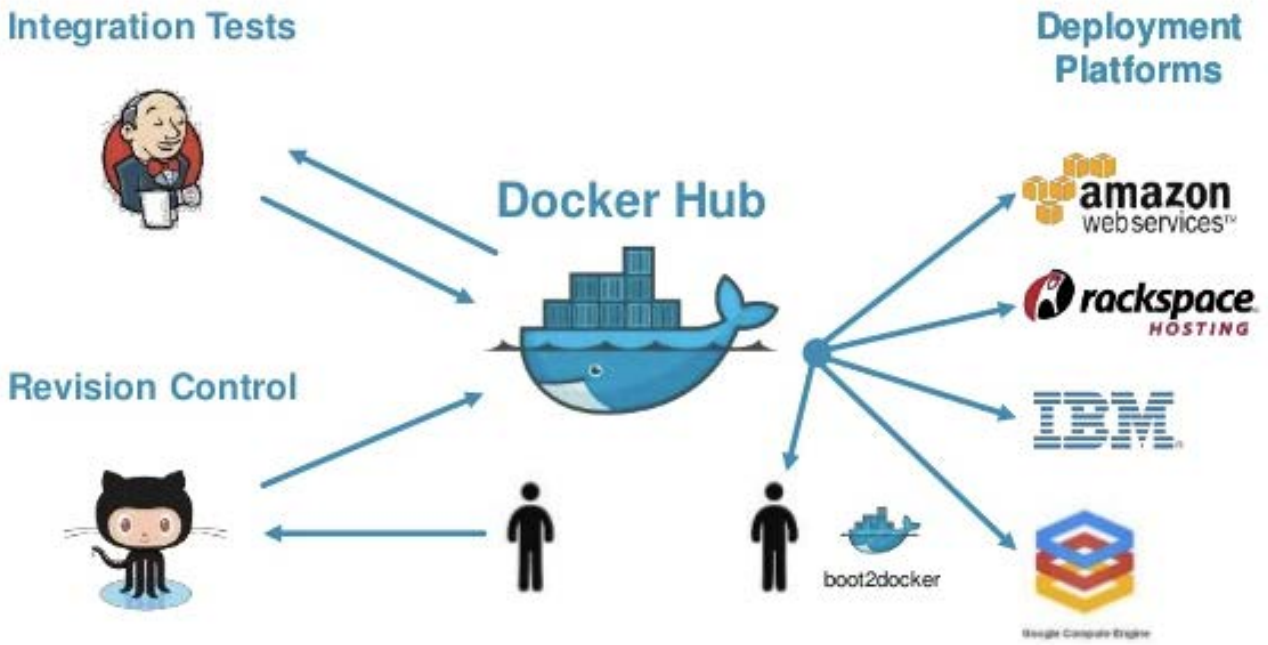
- Root namespace >> ubuntu:trusty
- User namespace >> devuser/image:version
- Self-hosted namespace >> regsitry-server:5000/imagename:version

The dockerfile

- Provides a set of instructions for building docker images
- Creates a repeatable process
- Uses a local cache system for rapid builds

SHOW ME AN EXAMPLE

CI/CD workflow with Docker



Registries

- Host images
- Automate workflow
- Provide authentication
- Allow collaboration

The new artifact

VS

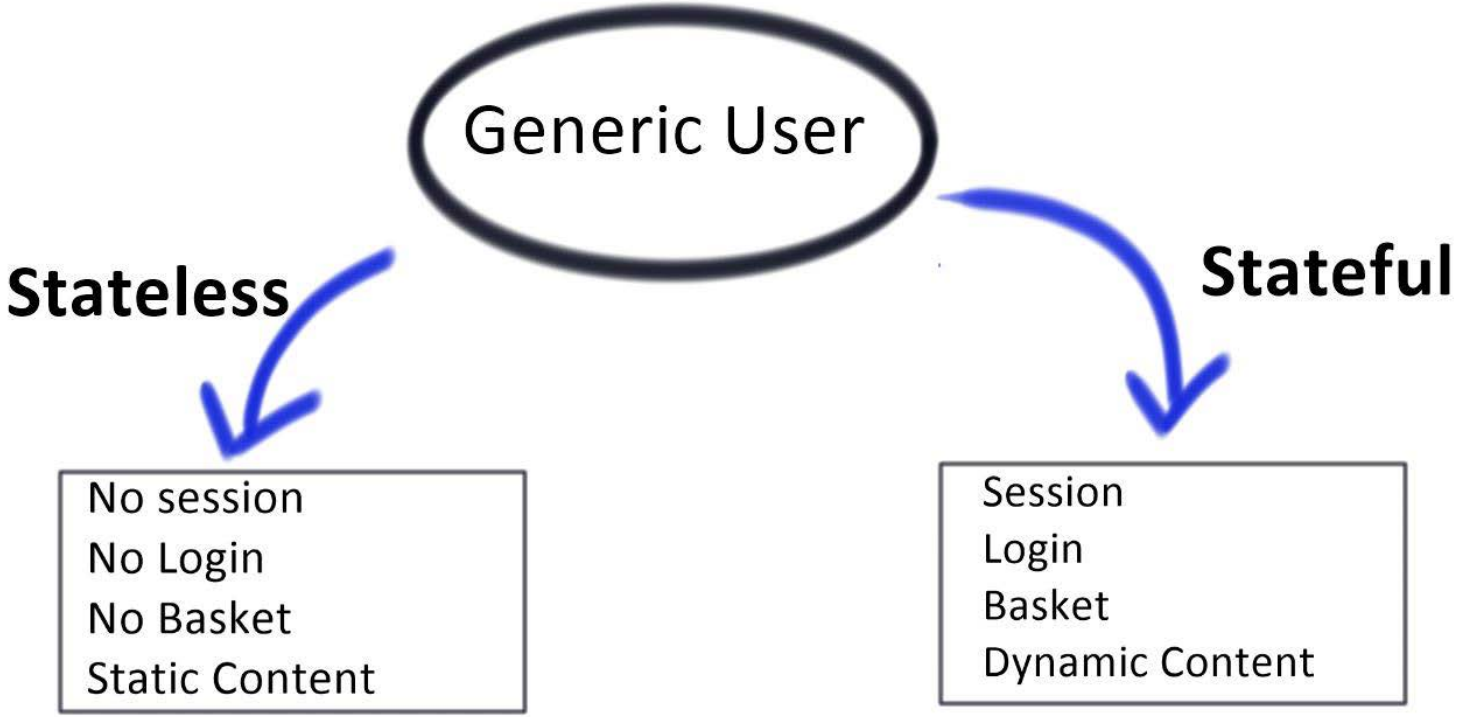
DEMO TIME

Use case | unit testing

- Continuous integration
- Density of containers
- Speed of provisioning
- Integration with existing CI servers

DEMO TIME

State vs Stateless

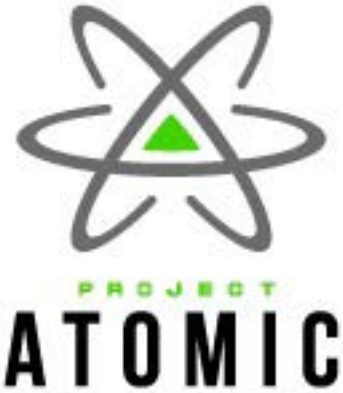


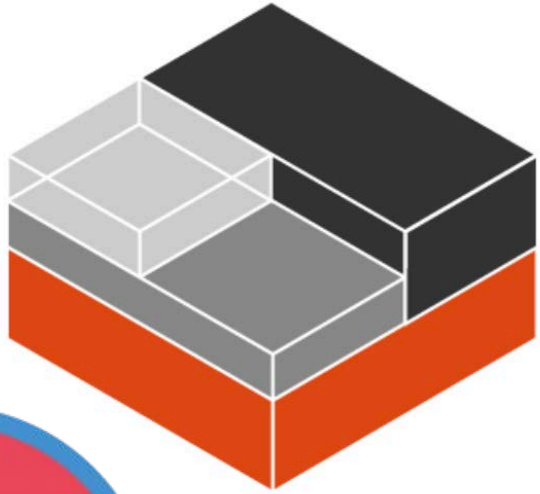
Containers in action

The Unicorns

- Are born in the Cloud
- Deliver a common solution
- Everyone “gets it”

Containers as an OS





The challenges

- Networking – redefining the network control plane
- Stateful Apps – maintaining state in an app
- Monitoring – chaining how systems tools monitor isolated ephemeral process
- Orchestration – managing and scaling instances in real time

The network between hosts

- Control plane in the App
- Extending a label's scope
- Performance costs

Achieving State

- Copy on Write
- 2 methods of access
  - Graph drivers
  - Volume mounts

Graph Drivers

- AUFS – the original driver used
- Device Mapper – the Redhat default (lvm2)
- Btrfs – Upstream on Fedora
- Overlayfs – the future maybe?
- VFS – when nothing else is available
- ZFS – experimental support

Volume mounts

- Interface
  - Mount
  - Unmount
  - Create
  - Remove
  - Path
- Available via Volume command

Monitoring and Service discovery

- Provide configuration to new containers
- Update other services to use new container
- Provide interface for monitoring service

Orchestration

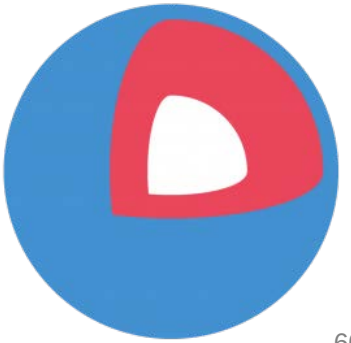
- 2 types of orchestration
  - Placement orchestration
  - Dynamic orchestration
- Docker’s take

Placement orchestration

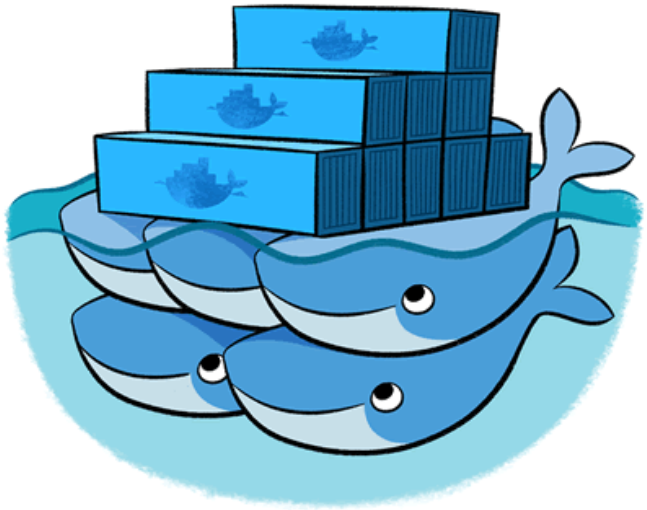
- New Relic and Spotify
- Developed as part of a DevOps process
- Actively used in production
- OSS via github

Dynamic Orchestration

- Ability to scale containers
- Respond to load
- Facilitate networking of containers
- Provide garbage collection
- Provide logging facilities



Docker's Take



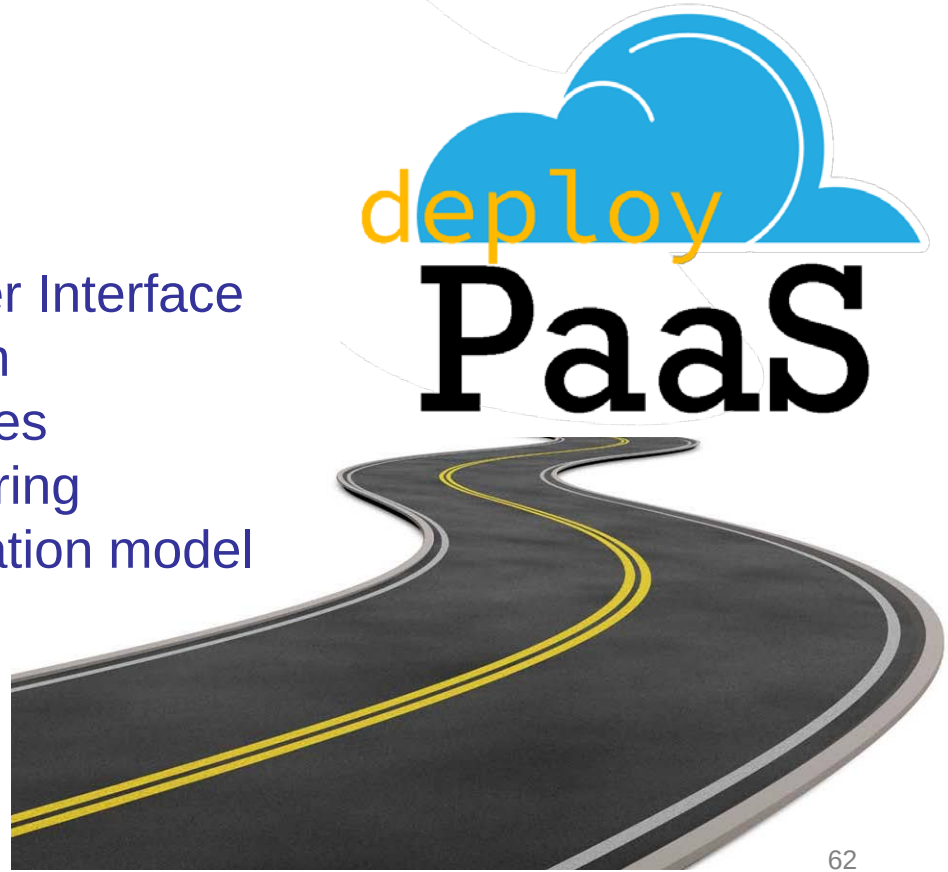
Docker Swarm

- Compatible with Docker Tools
- Pluggable Schedulers
- Smart Container Scheduling.
- Pluggable Node Discovery

Is this the road to the PaaS

PaaS Characteristics

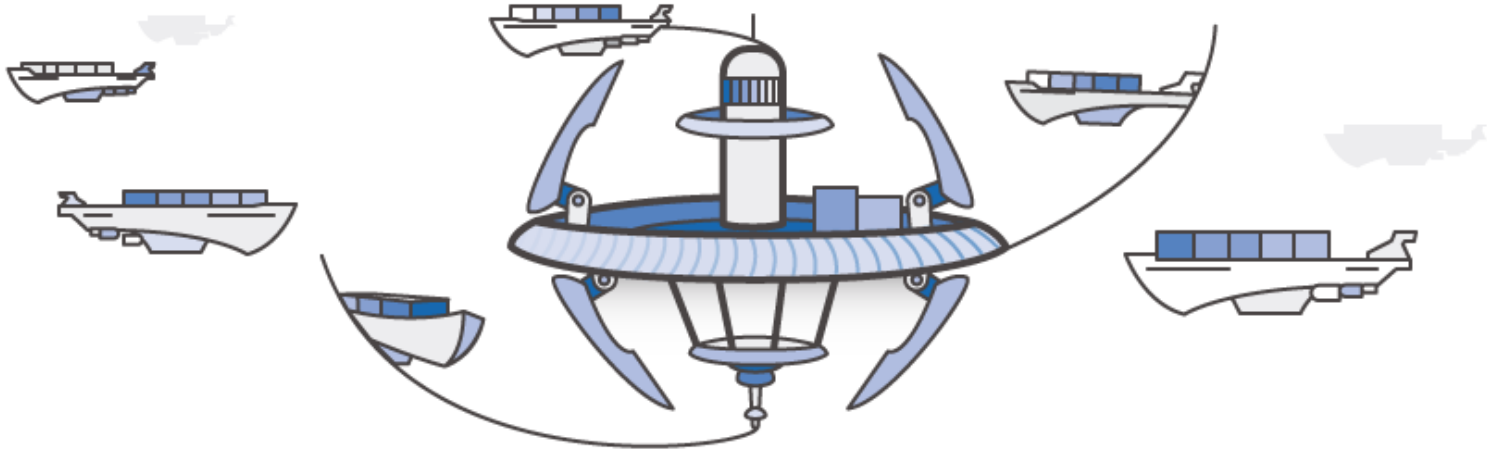
- Multi-tenant architecture
- Customizable /Programmable User Interface
- Unlimited Database Customization
- Robust Workflow engine/capabilities
- Granular control over security/sharing
- Flexible “services-enabled” integration model



Who's cloud?

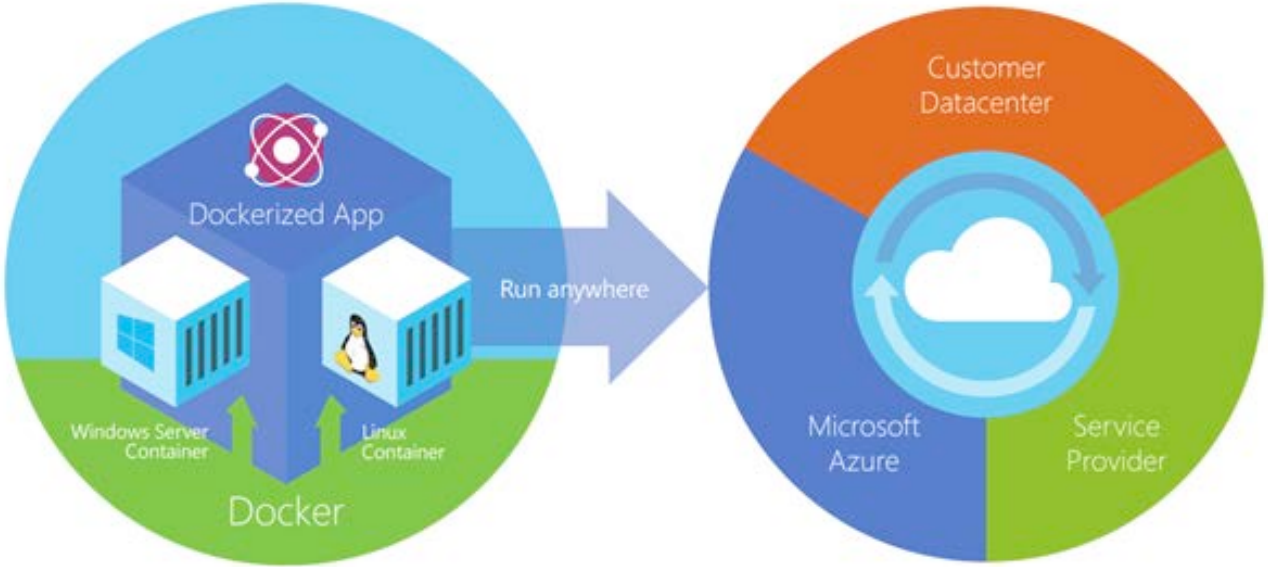
- Hybrid
- The heterogeneous container cloud
- The Players
  - AWS | Azure | Google Compute
  - Digital Ocean | Joyent | Others

AWS - ECS



Highly scalable, high performance container management

Azure


Dockerized App
Windows Server Container
Linux Container
Docker
Run anywhere
Customer Datacenter
Microsoft Azure
Service Provider

Google Container Engine

- Docker support
- Better ops
- Declarative management
- Scalable
- Cloud flexibility
- Easy container organization

Future Trends

A Docker Alternative

- New and Old runtimes
- System vs App Containers
- Hypervisor of containers
- AppC vs Open Container

Security in a Multitenant Model

- Guaranteed images
- VM like containers
- Role based security

Commoditization

Compute as a commodity…
Leads to containers everywhere!

Conclusion

Docker information resources

- Docker homepage - http://www.docker.com/
- Docker Hub - https://hub.docker.com
- Docker blog - http://blog.docker.com/
- Docker documentation - http://docs.docker.com/
- Docker Getting Started Guide - http://www.docker.com/ gettingstarted
- Docker code on GitHub - https://github.com/docker/docker
- Docker mailing list - https://groups.google.com/forum/#!forum/docker-user
- Docker on IRC: irc.freenode.net and channels #docker and #docker-dev
- Docker on Twitter - http://twitter.com/docker
- Get Docker help on Stack Overflow – http://stackoverflow.com/search?q=docker

Additional Information

- http://blog.scottlowe.org/2013/11/25/a-brief-introduction-to-linux-containers-with-lxc/
- http://www.linuxjournal.com/content/linux-containers-and-future-cloud?page=0,2
- https://access.redhat.com/articles/1353593
- https://blog.engineyard.com/2015/isolation-linux-containers
- http://www.toptal.com/linux/separation-anxiety-isolating-your-system-with-linux-namespaces
- http://developerblog.redhat.com/2014/09/30/overview-storage-scalability-docker/
- https://labs.ctl.io/how-to-use-different-docker-filesystem-backends/
- http://blog.emccode.com/2015/06/22/volume-drivers-arrive-at-dockcon/
- http://blog.weave.works/2015/07/14/i-quadrupled-the-speed-of-our-container-test-rig-in-a-day/
- http://blog.thestateofme.com/2015/03/09/using-overlay-file-system-with-docker-on-ubuntu/
- http://qconlondon.com/system/files/presentation-slides/Docker%20Clustering.pdf
- http://jpetazzo.github.io/assets/2015-07-01-deep-dive-into-docker-storage-drivers.html#48
- https://jpetazzo.github.io/2014/01/29/docker-device-mapper-resize/

Thank You