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

2006 – Google “Control Groups”

2007/8 – Merged into kernel 2.6.24

2008 – LXC project adds namespaces

2013 – namespaces redesign merged into kernel 3.16

2013 – lmctfy project create by Google
VM vs Containers

VM Diagram:
- App 1: Bins/Libs, Guest OS
- App 2: Bins/Libs, Guest OS
- App 3: Bins/Libs, Guest OS

Containers Diagram:
- App 1: Bins/Libs
- App 2: Bins/Libs
- App 3: Bins/Libs

- Docker Engine
- Operating System
- Infrastructure

Flash Memory Summit 2015
Santa Clara, CA
Benefits of Process Virtualization

- Density
- Provisioning
- Performance
Drawbacks of Process Virtualization

- Reduced isolation
- Reduced security
C-Groups

cgroups is merely a hierarchy of processes

• Resource limitation
• Prioritization
• Accounting
• Control
Namespaces

Process

Network

Mount

IPC

UTS

User
The Process namespace provides isolation for the allocation of process identifiers (PIDs).
The Network namespace isolates the network interface controllers (physical or virtual), iptables firewall rules, routing tables etc.
The Mount namespace allows the creation of a different file system layout, or making certain mount points read-only.
The IPC namespace isolates the System V inter-process communication between namespaces.
UTC Namespace

The "UTS" namespace allows changing the hostname.
The User namespace isolates the user IDs between namespaces.
Docker Architecture
Docker who?

Company | Project | Container
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAN-2013</td>
<td>dotCloud starts Docker project</td>
</tr>
<tr>
<td>AUG-2013</td>
<td>eBay, Uber, Cloudflare start using Docker</td>
</tr>
<tr>
<td>OCT-2013</td>
<td>Docker, Inc. splits from dotCloud</td>
</tr>
<tr>
<td>DEC-2013</td>
<td>Google CE announces support for Docker</td>
</tr>
<tr>
<td>APR-2014</td>
<td>AWS announces support for Docker</td>
</tr>
<tr>
<td>JUN-2014</td>
<td>Docker 1.0 released @ dockercon</td>
</tr>
<tr>
<td>SEP-2014</td>
<td>$40 Million – C round funding</td>
</tr>
<tr>
<td>OCT-2014</td>
<td>Partnership with Microsoft announced</td>
</tr>
<tr>
<td>DEC-2014</td>
<td>Docker announces Machine, Compose, Swarm</td>
</tr>
</tbody>
</table>
The Challenge
Finding Inspiration

450,000 → 45,000
Container for Apps

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.

Flash Memory Summit 2015
Santa Clara, CA
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

Any App
+ 45K apps
+ 16K projects

API

Engine
open source software at the heart of the Docker platform

Hub
cloud-based platform services for distributed applications

Any infrastructure
• Physical
• Virtual cloud

API
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
How to get Docker?

- Linux installers
- Script from docker > [http://get.docker.com](http://get.docker.com)
- Get the binary > [https://get.docker.com/builds/Linux/x86_64/docker-latest](https://get.docker.com/builds/Linux/x86_64/docker-latest)
- Boot2docker or Kitematic
Boot2docker vs Kitematic
DEMO TIME
Containers enabling CI/CD
Docker Image format
Easy to Ship
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 >> registry-server:5000/imagenname: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
DEMO TIME
Use case | unit testing

- Continuous integration
- Density of containers
- Speed of provisioning
- Integration with existing CI servers
DEMO TIME
State vs Stateless

Generic User

Stateless
No session
No Login
No Basket
Static Content

Stateful
Session
Login
Basket
Dynamic Content
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
• 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
Highly scalable, high performance container management
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://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://developerblog.redhat.com/2014/09/30/overview-storage-scalability-docker/
- https://labs.ctl.io/how-to-use-different-docker-filesystem-backends/
- http://jpetazzo.github.io/assets/2015-07-01-deep-dive-into-docker-storage-drivers.html#48
Thank You