WHAT’S REALLY THE DIFFERENCE

Between a VM and a Container?
ONE SIMPLE IDEA

CHANGED EVERYTHING.
1873
1896
WE COULDN'T IMPROVE THE PRODUCT SO WE IMPROVED THE TUBE.

- Colgate, 1908
1978

THE LAB ASSISTANT
OMG, SALES DOUBLED!
WELL, ACTUALLY...
TOOTHPASTE TUBE THEORY

1) PRESSURE BUILT UP IN A FINITE BOUNDED SYSTEM NEEDS TO BE RELEASED SOMEWHERE OR THE SYSTEM WILL BREAK.

2) THERE ARE DIMINISHING RETURNS TO SQUEEZING THE TUBE AFTER A CERTAIN POINT.
ADRIAN OTTO

Distinguished Architect, Rackspace
Founder, OpenStack Containers Team
Founder and PTL, OpenStack Magnum
Organizer, Docker Los Angeles
THE DIFFERENCE

1. EFFICIENCY
2. PERFORMANCE
3. SECURITY
• 1960’s IBM S/360 Mainframes are the 800# Gorilla
  • Single user system designed for batch jobs
• 1963 MIT Project MAC ($2M grant from DARPA)
  • MAC = Multiple Access Computing: Multics
  • Vendor Choice == GE (Commercial interest in time sharing computer)
  • Whoops! IBM panicked! Created CP-40 for Bell Labs, CP-67.
    • Virtual Machines on the CP-67 using “CP (Control Program)” in 1967!
• 1987 Insignia Solutions “SoftPC”
• 1997 Apple (Connectrix) “VirtualPC”
• 1999 VMWare “VMWare Workstation”
APPLICATION VIRTUALIZATION

- 1990 Sun Microsystems “Stealth”
  - Address C/C++ Portability problems
  - Renamed Oak -> Webrunner -> Java (1995)
- 1996 Sun Microsystems “Java”
  - Java Development Kit (JDK)
  - Java Runtime Environment (JRE)
  - Java Virtual Machine (JVM)
OPEN SOURCE VIRTUALIZATION

Xensource
First open source hypervisor!

2003

2004

2005

2006

2007

Oracle VirtualBox OSE
(Open Source Edition)

Linux KVM
Kernel 2.6.20
HISTORY OF CONTAINERS (1/2)

- 1979 UNIX chroot (added to BSD in 1982)
- 2000 FreeBSD Jails (filesystems, users, networks)
- 2001 Linux VServer (VPS Solution)
- 2005 OpenVZ (filesystems, users/groups, process tree, networks, devices, IPC)
- 2006 Process Containers (Linux Kernel 2.6.24, limit CPU, mem, disk, network IO)
- 2008 Control Groups (cgroups added to Linux Kernel)
- 2008 LXC (Linux Containers, CLI and language bindings for 6 languages)
- 2011 Warden, CloudFoundry
- 2013 LMCTFY, Google
HISTORY OF CONTAINERS (2/2)

- **2013**: DotCloud becomes Docker, Inc.
- **2014**: CoreOS introduces Rocket
- **2015**: Microsoft Windows Containers
EVERYTHING CHANGED IN 2013

2013

DOCKER IMAGE
LINUX CGROUPS

• Kernel Feature
• Groups of processes
• Control resource allocations
  • CPU
  • Memory
  • Disk
  • I/O
• May be nested
• Kernel Feature
• Restrict your view of the system
• Mounts (CLONE_NEWNS)
• UTS (CLONE_NEWUTS)
  • uname() output
• IPC (CLONE_NEWIPC)
• PID (CLONE_NEWPID)
• Networks (CLONE_NEWNET)
• User (CLONE_NEWUSER)
  • See also: privileged/unprivileged modes
• May be nested
• NOT A FILESYSTEM
• NOT A VHD
• Basically a tar file
• Has a hierarchy
• Arbitrary depth
• Layered filesystem
  • Top layer can be writable
• Fits into the Docker Registry
Git Repo Semantics
- Pull
- Push
- Commit
- Hierarchy
CONTAINER

• Combines several things
  • Linux Cgroups
  • Kernel Namespaces
  • Docker Image
  • Has a lifecycle
• Like a Makefile (shell script with keywords)
• Extends from a Base Image
• Results in a new Docker Image
• Imperative, not Declarative
FROM centos:centos6
MAINTAINER Adrian Otto <aotto@aotto.com>
RUN yum -y install httpd
EXPOSE 80
ADD start.sh /start.sh
CMD /start.sh

$ docker build -t webserver .
FROM webserver
MAINTAINER Adrian Otto <aotto@aotto.com>
RUN yum -y install mysql-server php
EXPOSE 80
ADD start.sh /start.sh
CMD /start.sh

$ docker build -t lampstack .
THE DIFFERENCE

1. EFFICIENCY
2. PERFORMANCE
3. SECURITY
# THE DIFFERENCE

## 1. EFFICIENCY

<table>
<thead>
<tr>
<th>App 1</th>
<th>App 2</th>
<th>App 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bins/Libs</td>
<td>Bins/Libs</td>
<td>Bins/Libs</td>
</tr>
</tbody>
</table>

### Operating System
- App 1
- App 2
- App 3

### Infrastructure
- Laptop
- Server
- Cloud

### Hypervisor

### Host Operating System

### Infrastructure
- Laptop
- Server
- Cloud
THE DIFFERENCE

2 PERFORMANCE

App 1
Bins/Libs

App 2
Bins/Libs

App 3
Bins/Libs

Operating System

Infrastructure

App 1
Guest OS

App 2
Guest OS

App 3
Guest OS

Host Operating System

Infrastructure

Hypervisor
THE DIFFERENCE

3 SECURITY

App 1
Bins/Libs

App 2
Bins/Libs

App 3
Bins/Libs

Operating System

Hypervisor

Host Operating System

Infrastructure

App 1
Guest OS

App 2
Guest OS

App 3
Guest OS

Infrastructure
CASTILLO DE SAN MARCOS
### VIRTUALIZATION MAPPINGS

<table>
<thead>
<tr>
<th>Physical</th>
<th>Virtual</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Partition</td>
</tr>
<tr>
<td>Logical Processor</td>
<td>Virtual Processor</td>
</tr>
<tr>
<td>Advanced Programmable Interrupt Controller (APIC)</td>
<td>Virtual APIC + Synthetic Interrupt Controller (SynIC)</td>
</tr>
<tr>
<td>Physical Address = System mPhysical Address (SPA)</td>
<td>Guest Physical Address (GPA)</td>
</tr>
</tbody>
</table>
LINUX SYSCALL INTERFACE

397 CALLS IN KERNEL 3.19
THE DIFFERENCE

SECURITY

39
CONTAINER ISOLATION TECHNIQUES

- SELinux / AppArmor
- Secure Computing Mode (seccomp)
- Container Nesting
- Docker Auth Plugins
- User Namespaces
- Encrypted Filesystems
- Address Space Layout Randomization (ASLR)
- Hardware Security Features (NX, VT-d, TPM, TXT, SMAP)
THE DIFFERENCE

1. EFFICIENCY
2. PERFORMANCE
3. SECURITY