Linux Containers Roadmap
Red Hat Enterprise Linux 7 RC

Bhavna Sarathy
Senior Technology Product Manager, Red Hat

Linda Wang
Senior Eng. Manager, Red Hat

Bob Kozdemba
Principal Soln. Architect, Red Hat
Defining the problem space

How do you deploy this service?

400 users

MongoDB Service

1 quad core

Server
Defining the problem space

How do you deploy this service?

400,000 users

Server w/ 4000 cores

MongoDB Services
Containerize!
How about Red Hat Linux Containers?

Value Proposition

Application isolation mechanism for Light-weight multi-tenancy

400,000 users

Server 1: 100 containers

Server 2: 100 containers

Server 40: 100 containers
RHEL 7 Linux Containers Use Case 1
Host Containers

- RHEL 7 host carved into secure containers
- Each container running RHEL 7 userspace
- Pro: Security errata can be applied easily with “yum update”
- Con: Limited to RHEL 7 runtimes
RHEL 7 Linux Containers Use Case 2
Image-based Containers

- Docker builds on Linux Containers and provides format for content distribution
- Docker includes the userspace runtime of an application
Image-based Containers with Docker technology

App Layer

Base Image

Layered Image 1

Layered Image 2

RHEL 7 Container Host

App A
SCL 1
RHEL 7 Runtime

App B
SCL 2
RHEL 6.5 Runtime

App C
SCL 3
RHEL 6.6 Runtime

Base Image

RHEL 7
September 19, 2013

RED HAT AND DOCKER COLLABORATE

We are thrilled to announce the collaboration between Docker and Red Hat.

Collaboration with Red Hat is important for a number of reasons, including:

- Driving compatibility with the most widely deployed Linux distributions
- Enabling integration with one of the most prominent and important PaaS solutions
- Collaborating with the most prominent, pure open source company

First, it is critically important for us to make Docker work seamlessly with Red Hat Enterprise Linux and related Linux distributions, such as Fedora. This is the #1 requested enhancement for Docker, and is obviously a major concern for people who want to deploy Docker in mainstream production environments. Our teams have been working together to package Docker for Fedora in time for the next release of Docker (0.7). Red Hat and dotCloud are planning to make Docker available for all Fedora users with upcoming releases, and we're making the initial package in a way that will ultimately be usable by Docker users.
Linux Containers in Red Hat Enterprise Linux 7

Identical Containers

Non-identical Containers

Host Containers

Image-based Containers

Red Hat Enterprise Linux 7

Host RHEL

RHEL 6
RHEL 7
RHSCL
Fedora
Containers can be deployed in baremetal or virtual.
RHEL 7 supports both Virtualization with KVM and Linux Containers.
RHEL 7 Linux Containers Benefits

- Integrated application delivery w/ Image-based solutions
- Application Mobility
- Light weight Application Isolation
- Minimal footprint

Simplified application delivery
RHEL 7 Container Host

- App A Runtime A
- App B Runtime B
- App C Runtime C

- Service Containers
- Mgmt Containers

- Container Host OS & Central Shared Services

- Kernel & HW Drivers

- Hardware
RHEL 7 Linux Containers - Building Blocks

Process Isolation

Resource Management

Management

Security
Linux Containers Features and Architecture
Linux Containers – Resource Management

Process Isolation

Resource Management

Security

Management
Resource Management – Control Groups

- CPU
- Memory
- Network
- Block IO

Cgroups

Linux Kernel

Hardware (Intel, AMD)
Control Groups – In a Container Environment
Control Groups – Resource Control
Control Groups - Usability Improvements

• RHEL6 Any privileged process can manage Cgroups
  • No coordination and with unexpected results
  • Kernel moving to single writer mode
  • Kernel does not enforce this yet...
• RHEL7 systemd will manage cgroups
  • Recommended to use systemd APIs in RHEL7
  • New concept of Scopes/Slices

https://www.youtube.com/watch?v=MSG4jW187Is
http://www.freedesktop.org/wiki/Software/systemd/ControlGroupInterface
Control Groups - Usability Improvements: Scopes

Systemd puts all related worker PIDs into cgroup called a ‘scope’.

• Services
  • Apache processes in same services/apache scope
  • Mysql processes in same services/Mysql scope
  • Apache/Mysql get an equal “slice” of the system

• Users accounts
  • All users get an equal “slice”

• Machines
  • All containers/VMs get an equal “slice”
  • No service/user/machine can dominate system
Control Groups – Systemd’s “Scope”
Control Groups – Systemd’s “Slice”
Control Groups - Usability Improvements: Slices

Special unit file for assigning resource constraints
Slices get assigned to scopes
• Systemd automatically assigns services to system.slice
• You can override resource with Unit file configuration
  • MemoryLimit=1g
• Command Line
  ```bash
  #> systemctl set-property httpd.service CPUShares=524 MemoryLimit=500M
  ```
• Systemd will assign Containers to machine.slice
  • You can override by editing
  • /etc/systemd/system/big-machine.slice
Linux Containers – Process Isolation

Process Isolation

Resource Management

Security Management
Process Isolation - Namespaces
Process Isolation - Namespaces

• Isolate processes
  • Create a new environment with a Subset of the resources
  • Once set up, namespaces are transparent for processes
• Can be used in custom and complex scenarios
• Supported Namespaces
  • ipc, pid, mnt, net, uts
  • Future Red Hat Enterprise Linux 7: user namespace
# Process Isolation - Namespaces

<table>
<thead>
<tr>
<th>Namespaces</th>
<th>Functionality</th>
<th>What does it mean?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount</td>
<td>Isolate the set of FS mount points seen by processes</td>
<td>/tmp in container can be different in ns’ Remount ‘/’ read only within namespace</td>
</tr>
<tr>
<td>PID (process ID)</td>
<td>Process can have same PID in different NS (include PID1)</td>
<td>Process in NS can’t see/interact with process outside</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All processes are visible in ‘root’ PID NS</td>
</tr>
<tr>
<td>Network</td>
<td>Isolate the networking stack: ip addr, routes, netfilter iptable rules</td>
<td>Each NS has its own private loopback IF Commonly used with virtual ethernet IF pair</td>
</tr>
<tr>
<td>UTS</td>
<td>Set a different host and domain names for NS</td>
<td>No impact to the rest of the system Useful when combined with Network NS</td>
</tr>
<tr>
<td>IPC</td>
<td>Private inter-process communication environment: message queues, semaphores,</td>
<td>Resources are only accessible within the Namespace</td>
</tr>
<tr>
<td></td>
<td>shared memory</td>
<td></td>
</tr>
</tbody>
</table>
Process Isolation - User Namespace

May be enabled in future releases...

- Mapping between UIDs and GIDs
  - 5000-6100 on Root Namespace
  - 0-1100 in User Namespace
- Super user (UID 0) possible inside the Namespace
  - Configuring network on a network namespace
  - Binding to ports < 1024
  - Treated as unprivileged UID 5000 outside namespace
Linux Containers - Security

Process Isolation

Resource Management

Management

Security
SELinux – Security
Each container process is confined in its own sandbox distinct from other the other processes.
Each process is confined in its own sandbox, distinct from the others.

When a process is attacked.....
...and compromised, there is far less exposure. Only the container process is lost – lose the process not the system.
We can label the sandboxes with a level of sensitivity and categories. Label the container process with multiple level using SELinux Multi Level Security (MLS)
SELinux and Docker

- Docker
- SELinux
- Docker Image
- Containers
  - Resource Mgmt
  - Namespaces
- RHEL Kernel
Linux Containers - Management

- Process Isolation
- Security
- Resource Management
- Management
System and Container Management
Docker CLI

• Tool to package an application and its runtime dependencies for deployment into a Linux Container
• Docker 0.9 includes libContainer, native LXC implementation
Red Hat Enterprise Linux 7 Containers
Architecture with Docker CLI

Containers

DOCKER CLI

SYSTEMD

Docker Image

Unit File

Cgroups
Namespaces
SELinux

Drivers

RHEL Kernel

Hardware (Intel, AMD)
Systemd Cgroup Configuration passed to Docker

Systemd

Docker

httpd

httpd_container.service
ExecStart: Docker start rhel7/httpd
MemLimit 500k

Unit File
Systemd Socket Activation of Docker Containers

**Systemd**

**httpd_container.service**

- ExecStart: Docker start rhel7/httpd
- [socket]
  - ListenStream=80

**Unit File**

Port 80
Systemd Socket Activation of Docker Containers

Systemd

Docker

httpd_container.service
ExecStart: Docker start rhel7/httpd
[socket]
ListenStream=80

Unit File
Red Hat Announces Certification for Containerized Applications, Extends Customer Confidence and Trust to the Cloud
Demo
Linux Containers in RHEL 7 - Key Takeways

• Application isolation mechanism for Light-weight multi-tenancy
• Application centric packaging w/ Docker image-based containers
• Linux Containers Productization
  • Key kernel enablers – full support in RHEL 7 GA
  • Docker 1.0 – shipped with RHEL 7 GA
• Linux Container Certification
• Red Hat and Docker partnership to build enterprise grade Docker containers
<table>
<thead>
<tr>
<th>Time</th>
<th>Title</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed 1:20 PM – 2:20 PM</td>
<td>Portable, lightweight and interoperable Docker containers across Red Hat Solutions</td>
<td>Room 236</td>
</tr>
<tr>
<td>Wed 3:50 PM – 5:50 PM</td>
<td>Containers &amp; resource management in Red Hat Enterprise Linux 7 Beta (Hand-on Lab)</td>
<td>Room 105</td>
</tr>
<tr>
<td>Wed 11:00 AM – 1:00 PM</td>
<td>Linux Containers and Application Isolation Demo</td>
<td>Red Hat Booth (Infrastructure Pod 1)</td>
</tr>
<tr>
<td>Wed 1:20 PM – 3:20 PM</td>
<td>Implementing &amp; managing OpenShift Enterprise</td>
<td>Labs II</td>
</tr>
<tr>
<td>Wed 11:30 AM – 2:00 PM</td>
<td>Next Generation Container Management Demo</td>
<td>Emerging Technologies Red Hat Booth (Pod 31)</td>
</tr>
</tbody>
</table>
Questions?