



redhat.

L103118 - LINUX CONTAINER INTERNALS

How they really work

Scott McCarty, Vinny Valdez, Jamie Duncan, Billy Holmes

Lab Gurus

5/4/2017

AGENDA

L103118 - Linux container internals

10:15AM—10:25AM

INTRODUCTION

10:25AM—10:40AM

ARCHITECTURE

10:40AM—11:05AM

CONTAINER IMAGES

11:05AM—11:35PM

CONTAINER HOSTS

11:35AM—12:05PM

CONTAINER ORCHESTRATION

12:05PM—12:15PM

CONCLUSION

Materials

The lab is made up of multiple documents and a GitHub repository

- Presentation (Google Presentation): <http://bit.ly/2pYAI9W>
- Lab Guide (this document): <http://bit.ly/2mIEIPG>
- Exercises (GitHub): <http://bit.ly/2n5NtPl>

CONTACT INFORMATION

We All Love Questions

- Jamie Duncan: @jamieeduncan jduncan@redhat.com
- Billy Holmes: @gonoph111 biholmes@redhat.com
- Vinny Valdez: @VinnyValdez vvaldez@redhat.com
- Scott McCarty: @fatherlinux smccarty@redhat.com

The background features a minimalist architectural design composed of large, overlapping geometric shapes. It includes a prominent red parallelogram on the left, a white triangle at the top, and several black and red rectangular blocks on the right. The overall aesthetic is clean and modern.

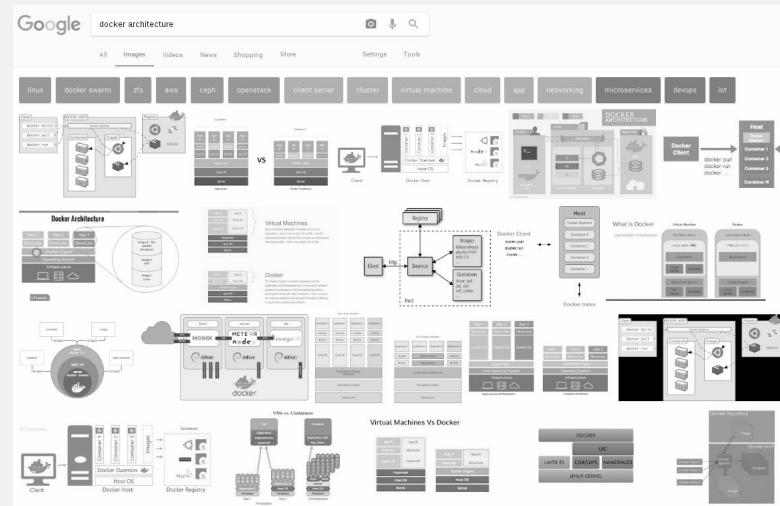
ARCHITECTURE

ARCHITECTURE

Google Images is Wrong :-)

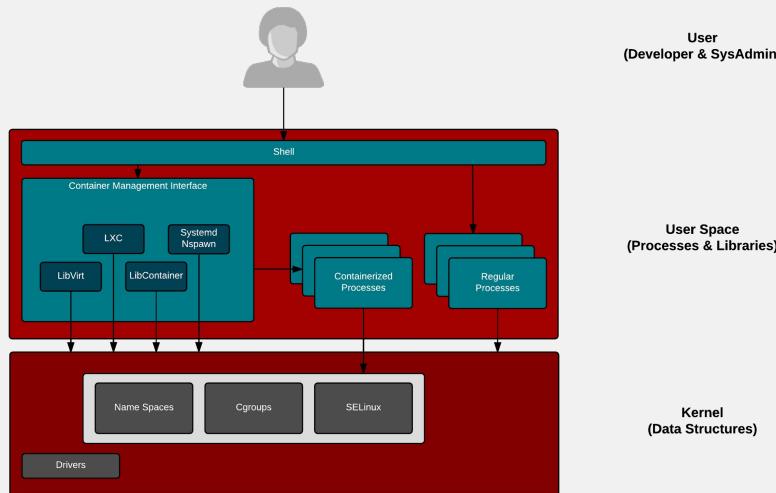
Important corrections

- Containers do not run ON docker.
Containers are processes - they run on the Linux kernel. Containers are Linux.
 - The docker daemon is one of the many user space tools/libraries that talks to the kernel to set up containers



CONTAINERS ARE LINUX

The Libraries, and Data Structures



Userspace libraries interact with the kernel to isolate processes

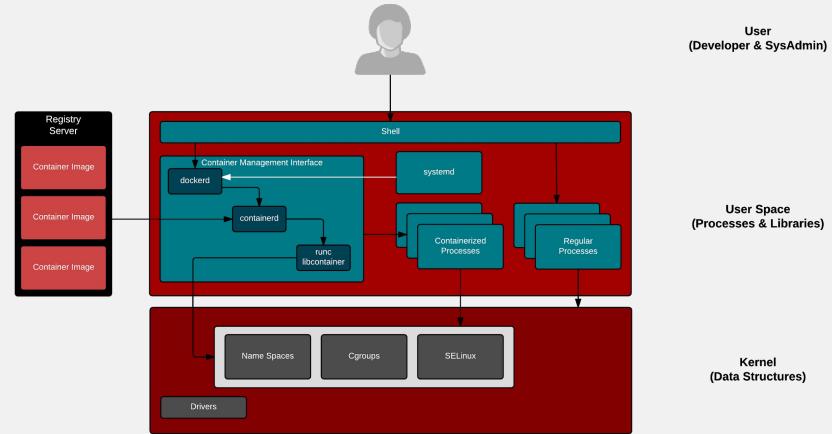
- Libraries
 - LXC, LXD, LibContainer, systemd nspawn, LibVirt
- Kernel Data Structures
 - Name Spaces
 - Cgroups
 - SELinux

THE USER SPACE TOOL CHAIN

On a Single Host

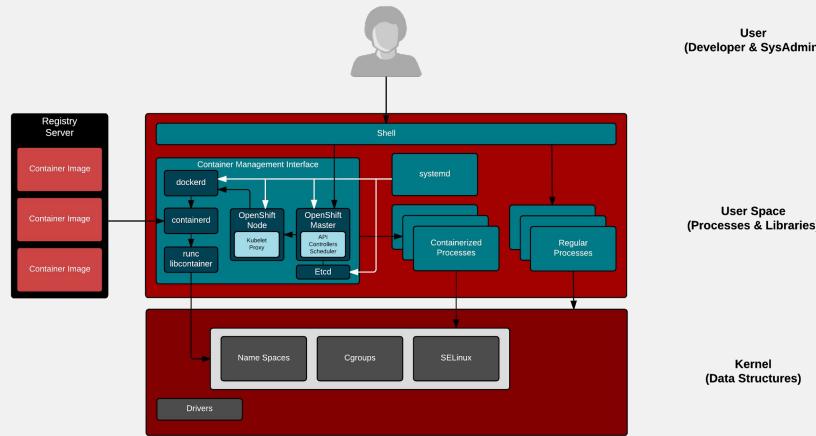
The user space tool chain adds the following:

- A local daemon
- Simple CLI/REST interface
- Support for container images (OCI) and connection to registries



THE ORCHESTRATION TOOLCHAIN

On Multiple Hosts



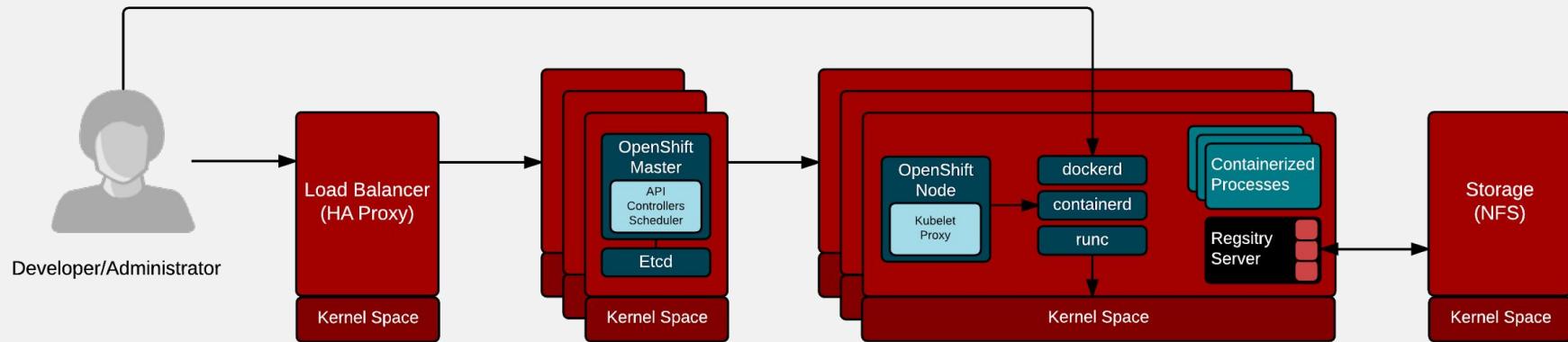
The orchestration toolchain adds the following:

- More daemons (it's a party) :-)
- Scheduling across multiple hosts
- Application Orchestration
- Distributed builds (OpenShift)
- Registry (OpenShift)

TYPICAL ARCHITECTURE

Bringing it All Together

In distributed systems, the user must interact through APIs

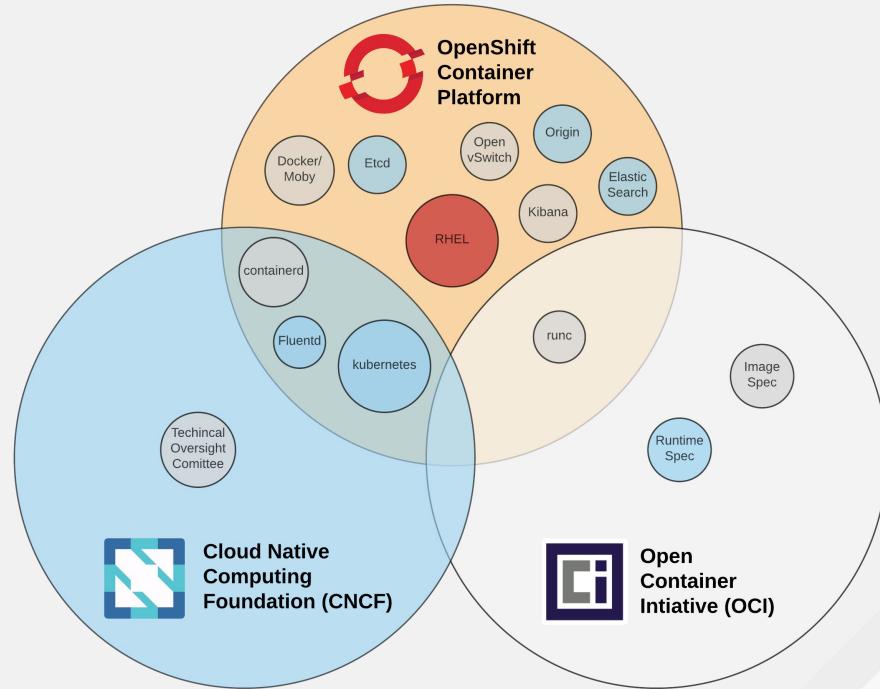


THE COMMUNITY LANDSCAPE

Open Source, Leadership & Standards

The landscape is made up of committees, standards bodies, and open source projects:

- Docker/Moby
- Kubernetes/OpenShift
- OCI Specifications
- Cloud Native Technical Leadership



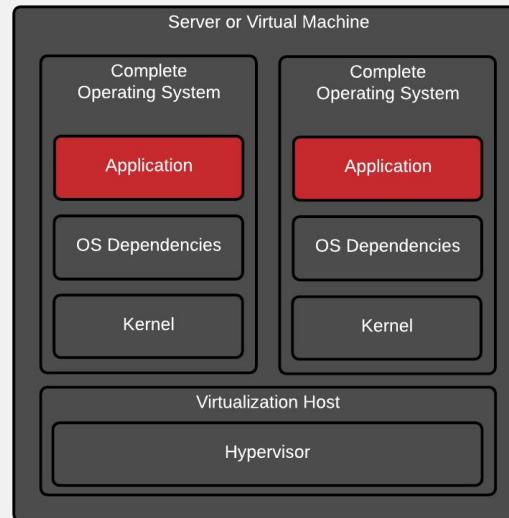
LAB 1: PAGE 3

A photograph of terraced rice fields in Sapa, Vietnam, showing the characteristic stepped patterns of the fields. A large, solid red diagonal shape covers the upper right portion of the image, partially obscuring the landscape.

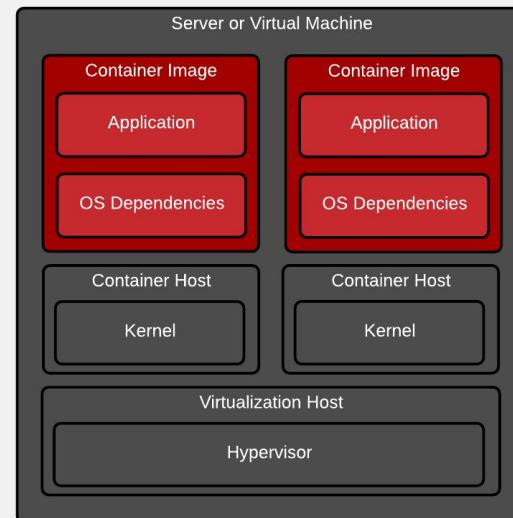
CONTAINER IMAGES

Container Images

Virtual machines and container environments



Application & Infrastructure
Updates Tightly Coupled



Application & Infrastructure
Updates Loosely Coupled

- Optimized for agility
- Optimized for stability

Fancy Files

People forget about Glibc...

User Programs

Library/Interpreter

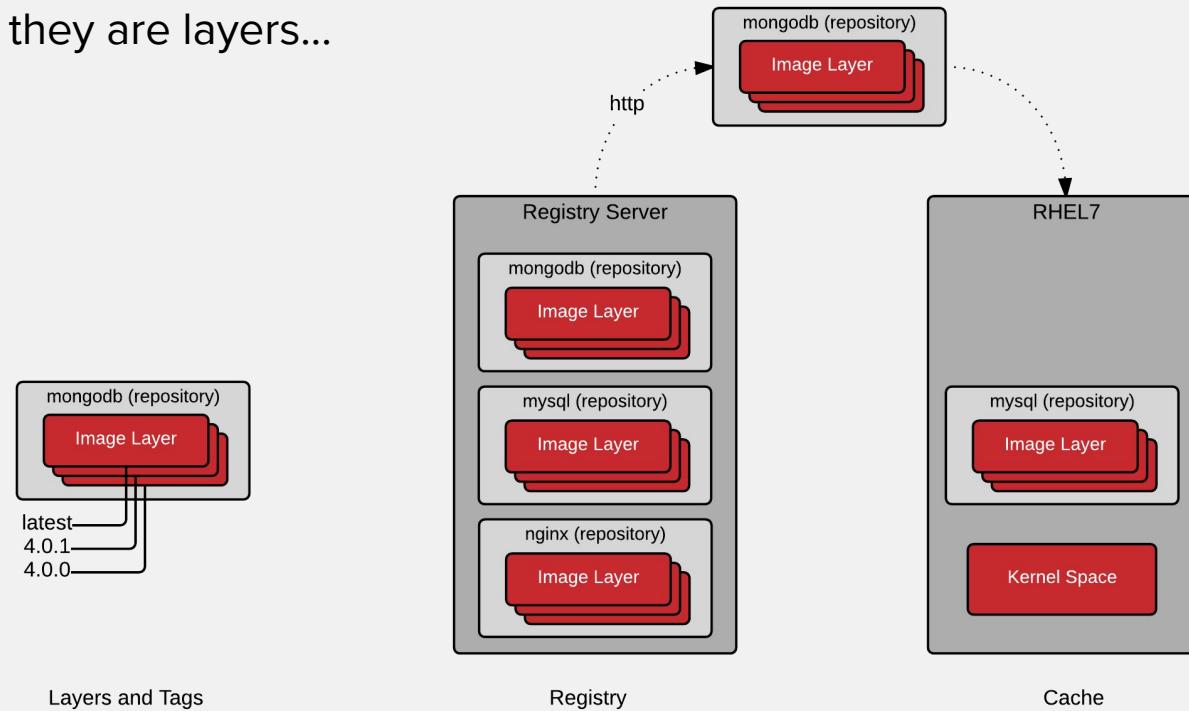
System Calls

Kernel Space



Fancy Files

Actually, they are layers...



Fancy File Servers

Actually, they are repositories

Command:

```
docker pull registry.access.redhat.com/rhel7/rhel:latest
```

Decomposition:

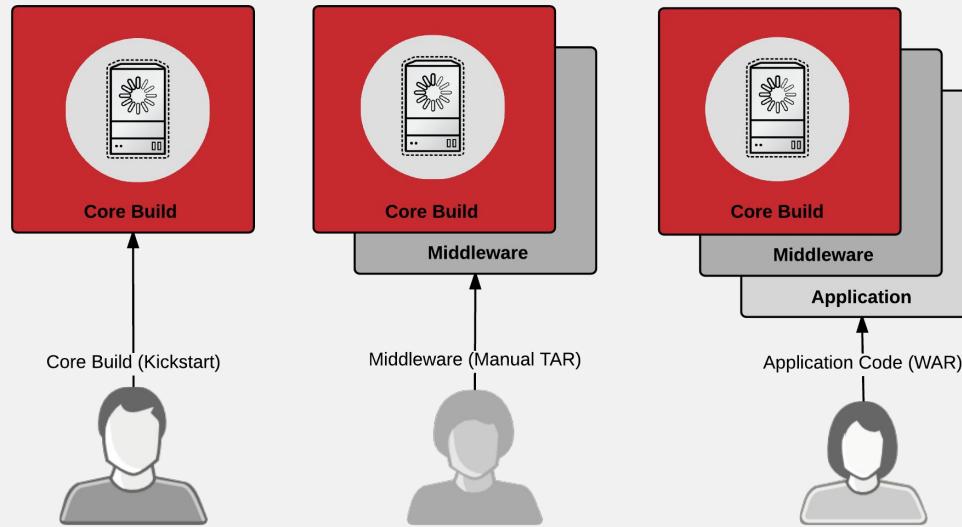
access.registry.redhat.com / rhel7 / rhel : latest

Generalization:

Registry Server / namespace / repo : tag

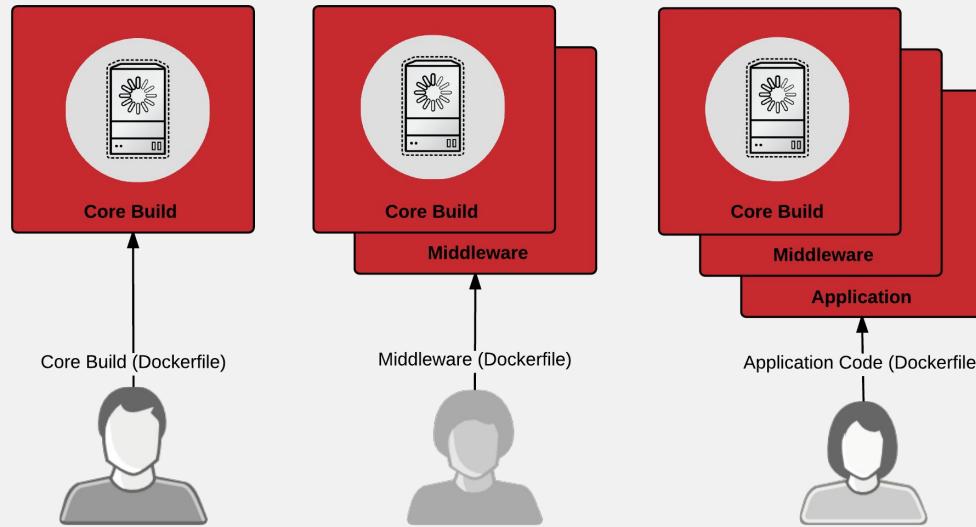
Fancy Files

How do we currently collaborate in the user space?



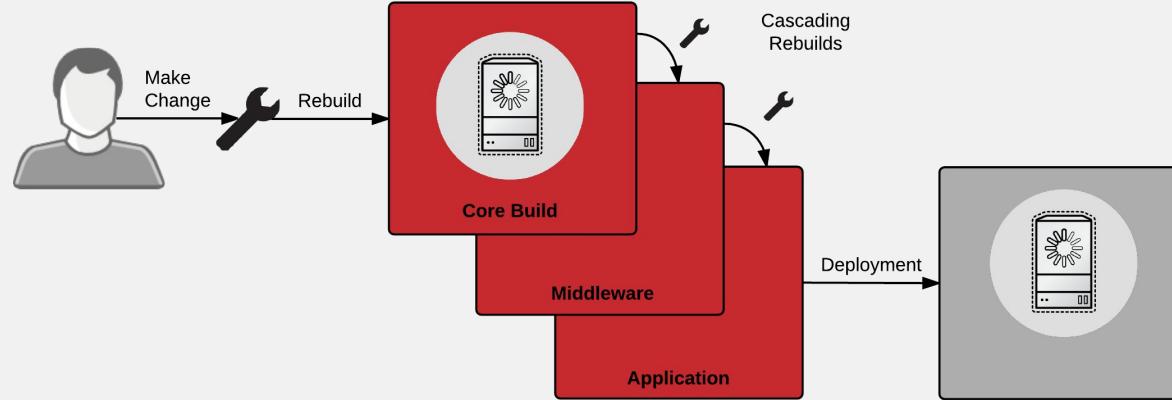
Fancy Files

The future of collaboration in the user space....



Fancy Files

The future of collaboration in the user space....



LAB 2: PAGE 8

CONTAINER RUNTIME

Fancy Processes

People forget about Glibc...

User Programs

Library/Interpreter

System Calls

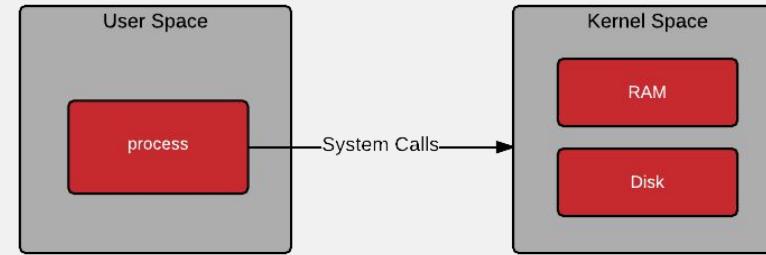
Kernel Space



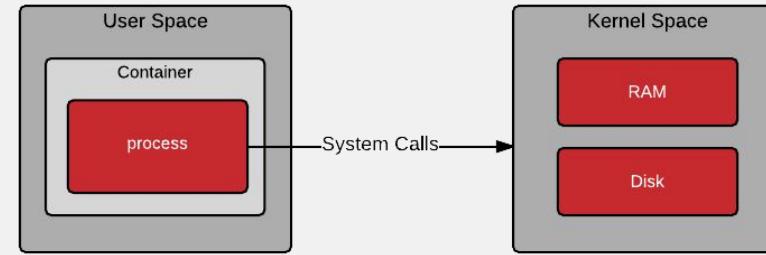
Linux Containers

Fancy Processes

Regular Linux Process

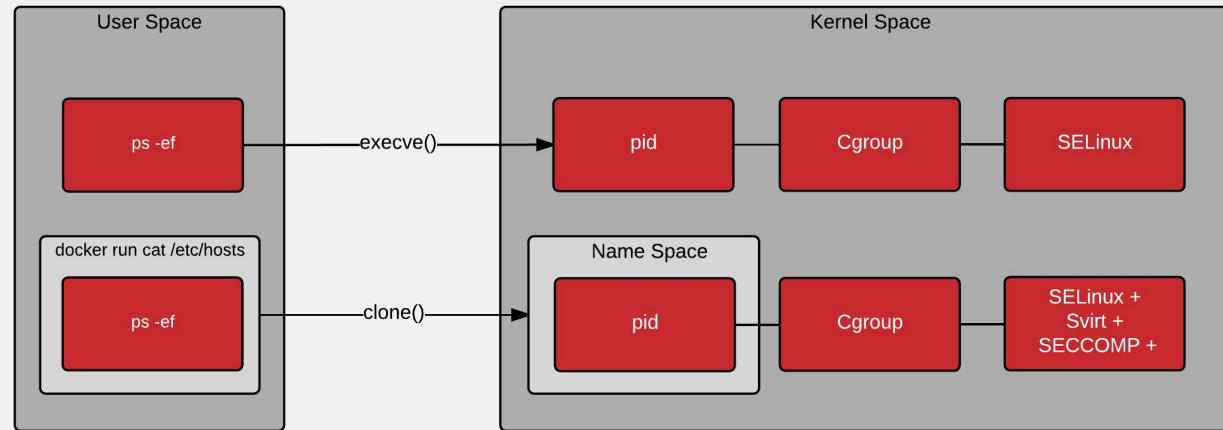


Containerized Process



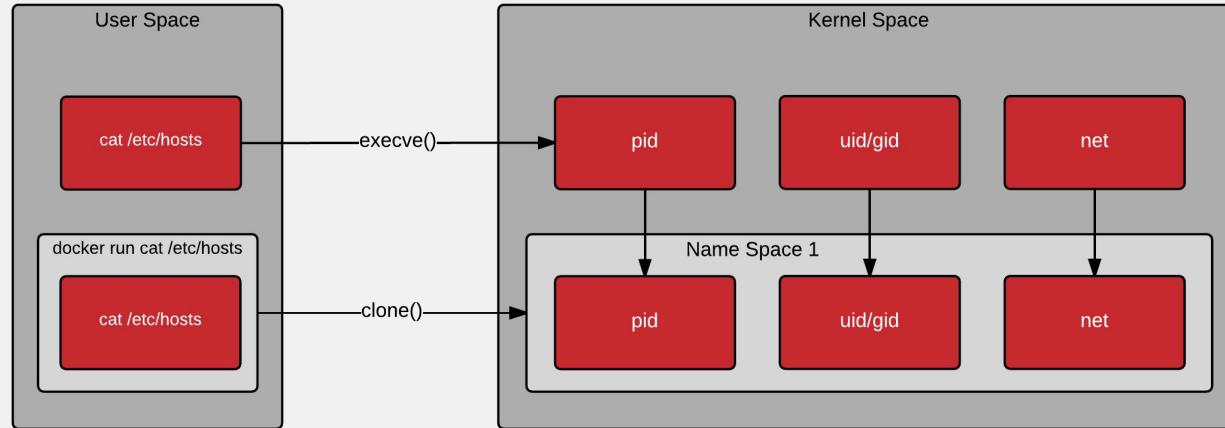
Containerized Processes

Starting the process with namespaces, cgroups, and security controls



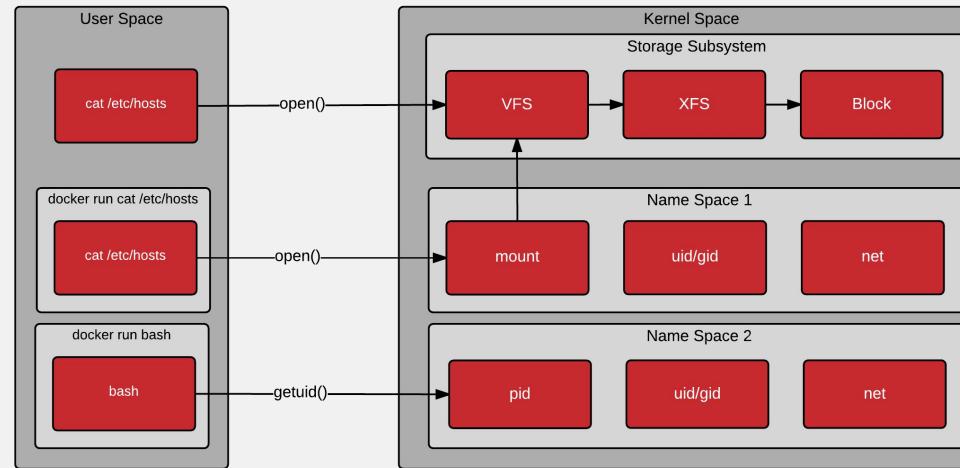
Containerized Processes

Starting the process in a namespace



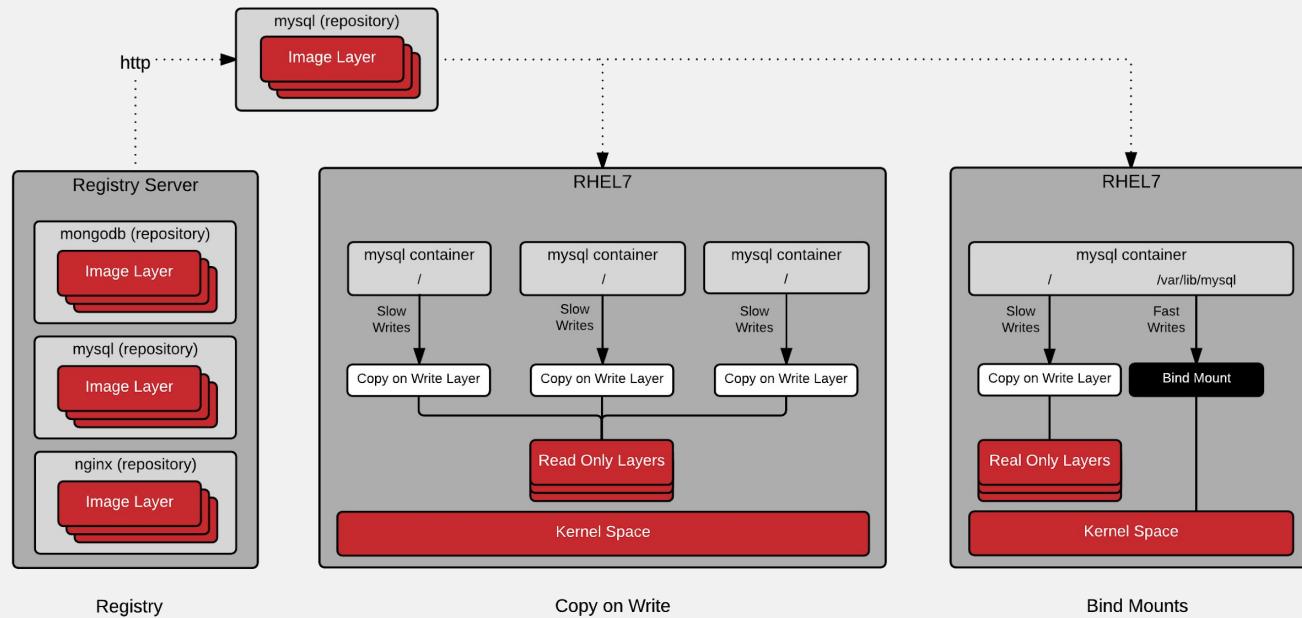
Containerized Processes

The containerized process still use the underlying kernel abstractions...



Mounts

Copy on write vs. bind mounts

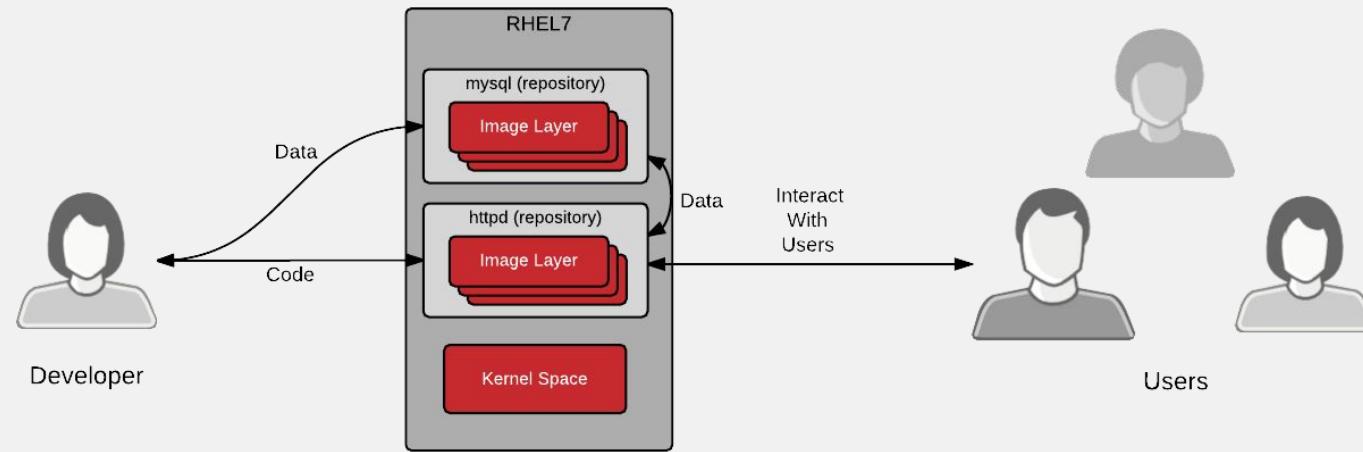


A large, semi-transparent red overlay covers the left side of the image, transitioning to a grayscale background on the right. In the background, a modern cable-stayed bridge with tall towers and multiple spans is visible against a clear sky.

CONTAINER ORCHESTRATION

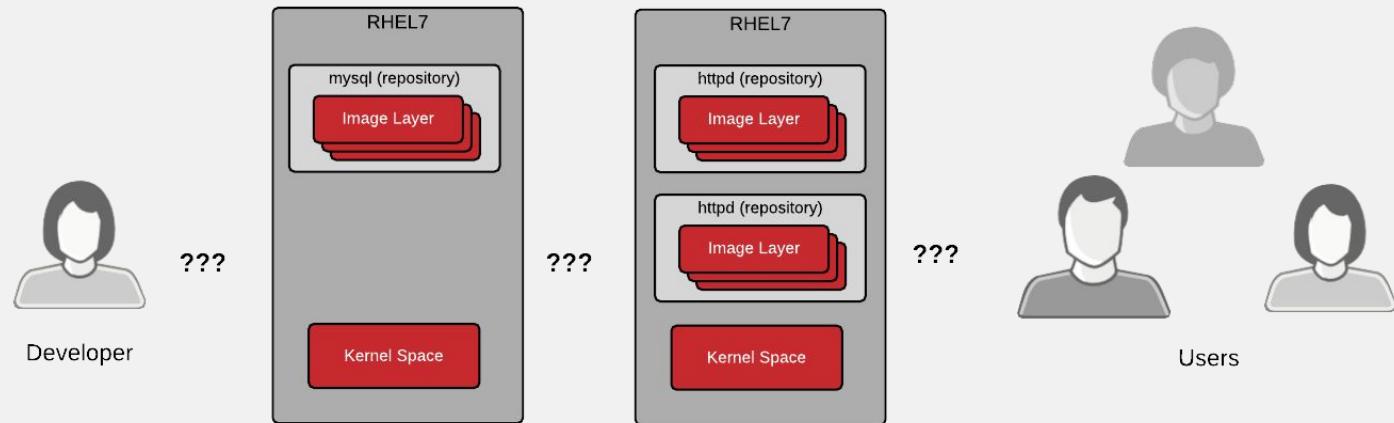
Application Containers

This is what most people think of with Docker



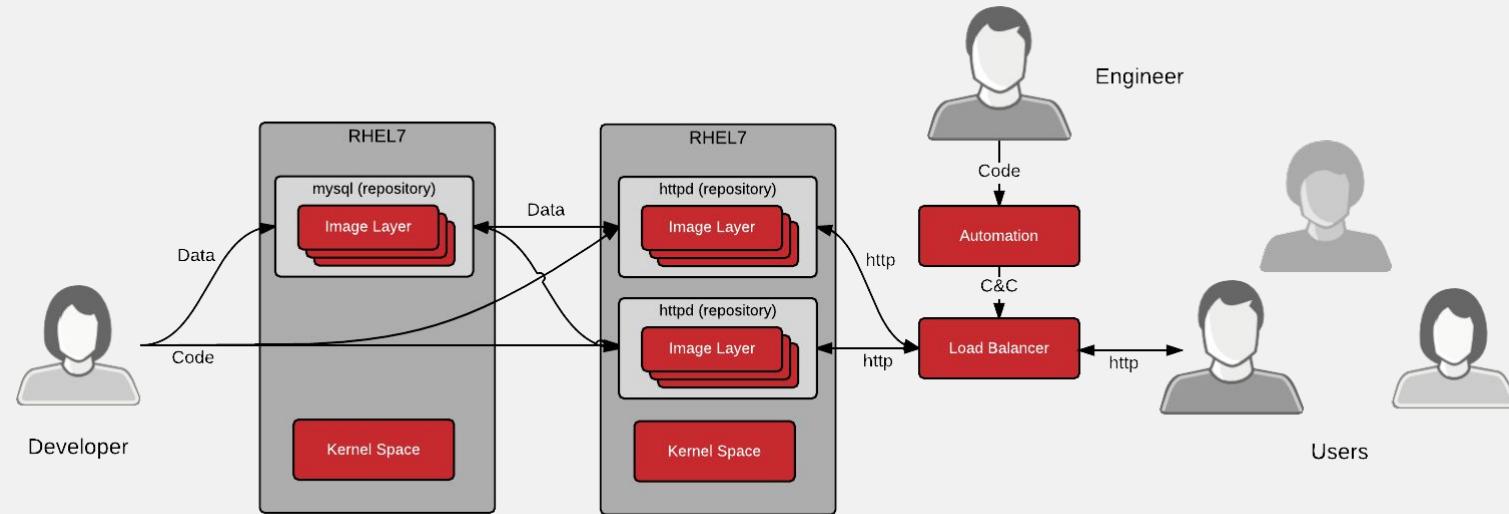
Container Orchestration

Multiple nodes changes everything



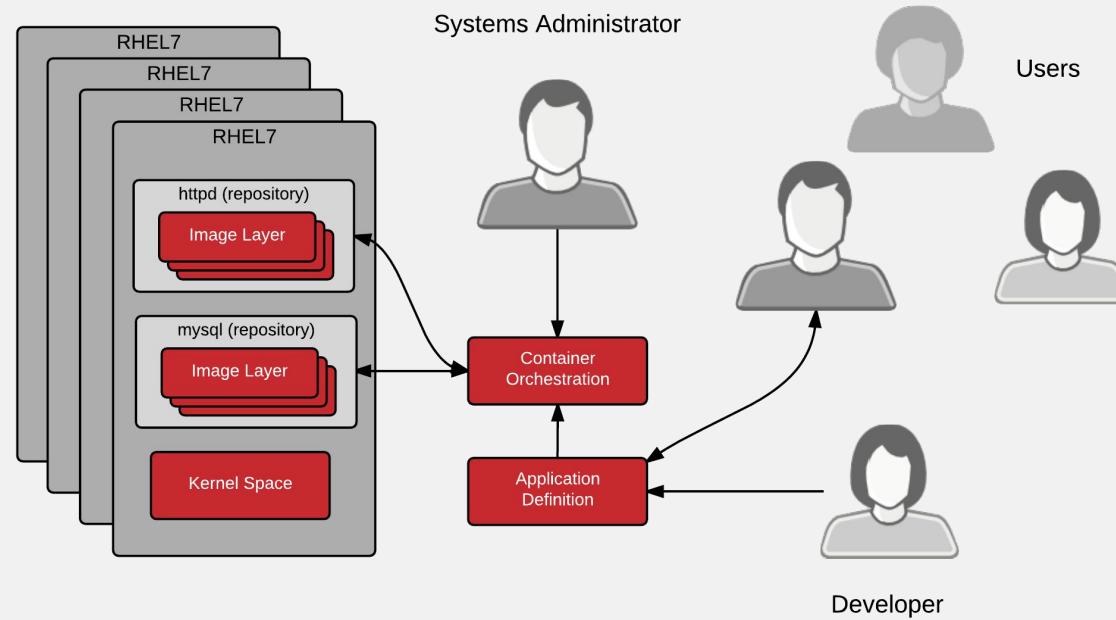
Container Orchestration

You can hack a solution together yourself, but it's ugly...



Kubernetes/OpenShift

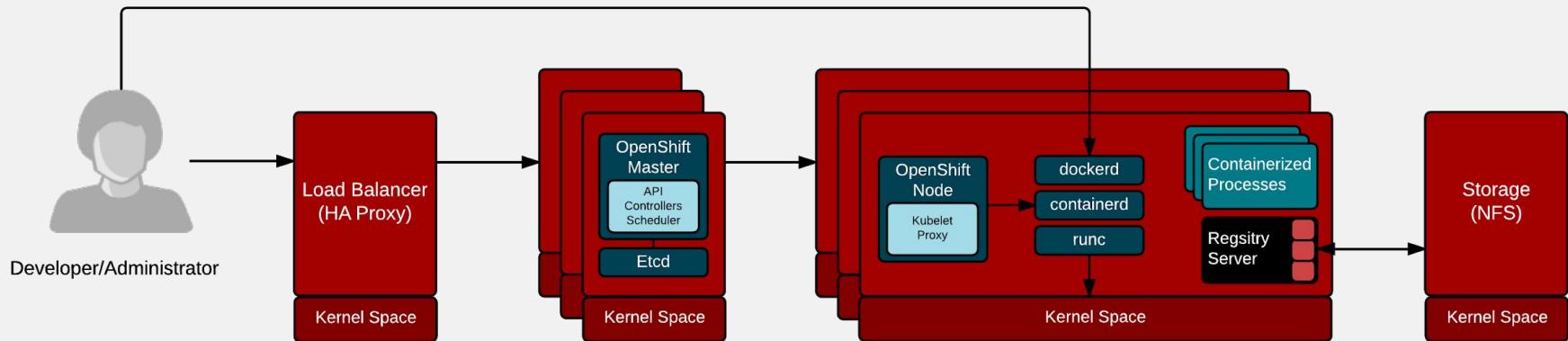
This Standardizes Everything



The Daemons

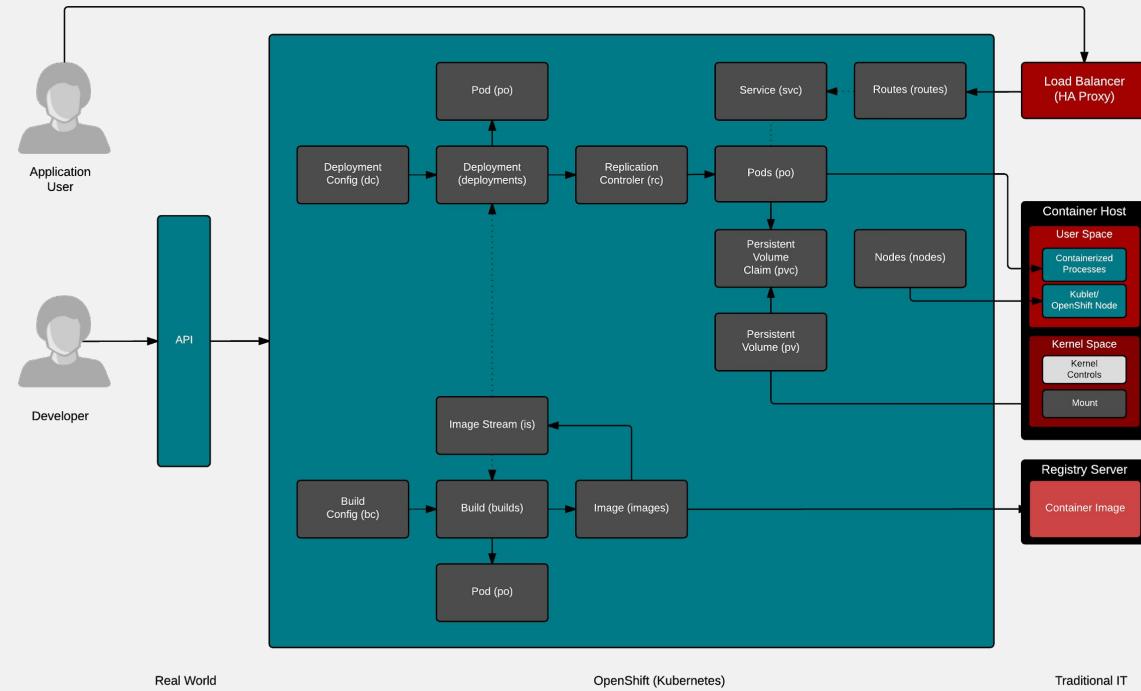
Bringing it All Together

User -> OpenShift -> Docker -> Kernel



THE LOGIC

Bringing it All Together





redhat.

THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos