

OpenShift on OpenStack in the MOC

Dan McPherson, Red Hat, Senior Principal Software Engineer

Rob Baron, Boston University, Senior Software Engineer

Containers and OpenStack

- What this talk **is not** about:
 - How running OpenStack in containers (Kubernetes) will solve all the world's problems (OSP 12)
- What this talk **is** about:
 - Real life experience of running OpenShift/Kubernetes on top of OpenStack in a education/research environment



MOC Introduction

- Cloud Computing environment backed by OpenStack
- Collaborative effort between 5 MA universities and multiple partners
- Provides an educational and research benefit
- Contrasts itself to Amazon/Microsoft/Google by wanting to have an open



Members

Academia and Nonprofit

Core Partners

The 'Members' section is divided into two categories: 'Academia and Nonprofit' and 'Core Partners'. The 'Academia and Nonprofit' section includes logos for Boston University, Harvard University, MIT, Northeastern University, UMASS, University of Massachusetts Lowell, Massachusetts Technology Collaborative, U.S. Air Force, and MOHPCC. The 'Core Partners' section includes logos for Brocade, Cisco, Intel, Lenovo, Red Hat, and Two Sigma.

MGHPCC

Massachusetts Green High
Performance Compute Center

~8000 sq meters

About the size of a Walmart

15 Megawatt

Can house 10,000 computers

LEED Platinum Certified

70% of the power is from a local



OPENSIFT

by Red Hat

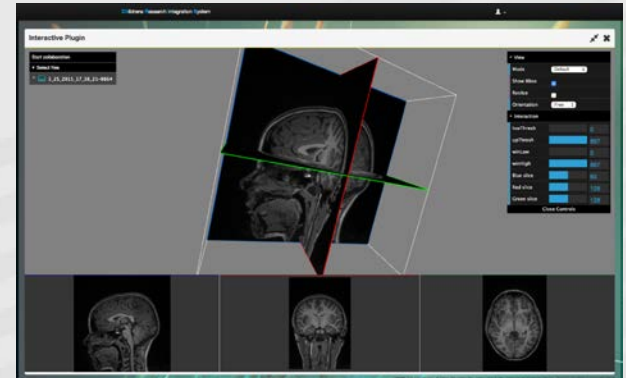
hydroelectric plant



Use Cases

Usage by the ChRIS Project

- ChRIS (Children's Research Integration System) is a web-based medical image platform that allows for various forms of medical image (Ex: MRIs) processing
- ChRIS itself is comprised of multiple open source projects (<https://github.com/FNNDSC/>) and the intention is to make the research and capabilities available to other hospitals



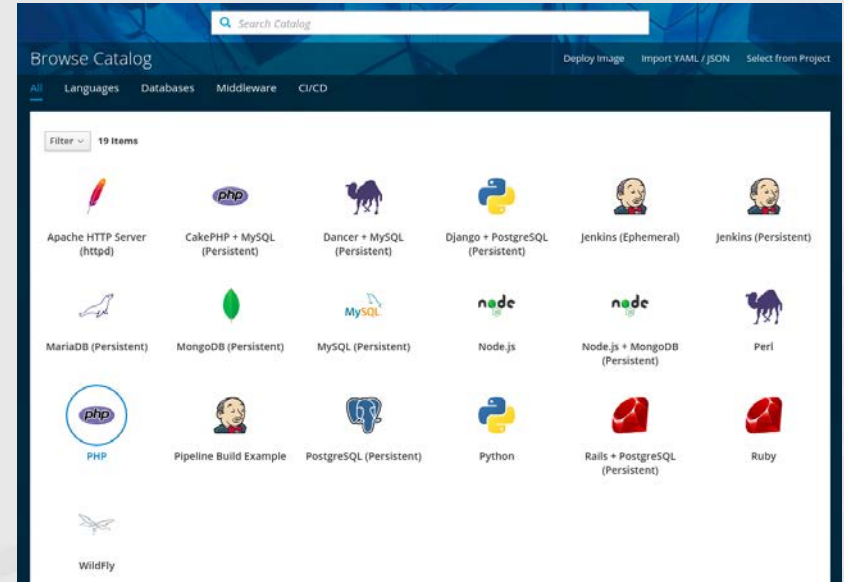
Benefits for ChRIS



- Using container images as a software distribution mechanism
 - Goal is to create a rich and shared ecosystem
 - Having a standardized platform is the key
- Scale out using OpenShift and OpenStack
- Will take advantage of passthrough GPU acceleration
- Potential to use slack resources with Kube Scheduler

Open Dataverse

- Use OpenShift / OpenStack as a means to run tools like Spark
- 20 petabyte data lake on the MOC
- Service Broker to connect to Dataverse



Students/Researchers

- Students/Personnel from 5 universities
- Common use cases include:
 - Course work
 - Website hosting
 - Research
 - Either related to OpenShift or simply using OpenShift



The Technology



OPENSIFT

What's OpenShift?

- A supported distribution of Kubernetes
- Adds developer and operational tooling on top of the base kubernetes distribution
- Available as a hosted platform, managed solution, and on-premises product



What's Kubernetes?

- A cluster orchestration tool for docker/OCI containers first introduced by Google in 2014
- By many measurements it's the most active project on GitHub
 - > 500 person years of work has gone into Kubernetes so far
- Red Hat is the #2 contributor

What's docker/OCI?



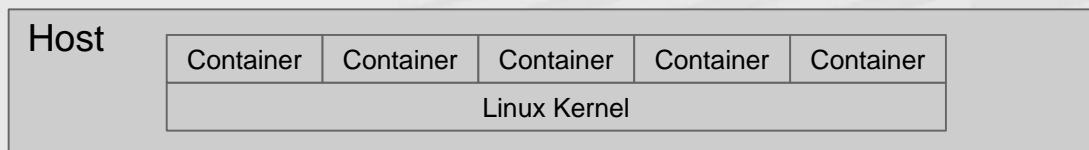
OPEN CONTAINER
INITIATIVE

- Docker is a container runtime and image format first launched in 2013
- OCI (Open Container Initiative - Est. 2015) now owns the specification which started as a de facto standard with docker

What's a Container?



- A Linux process wrapped with selinux, cgroups, and linux namespaces
- Provides multi-tenant isolation within a single host
- Adds near zero overhead compared to a virtual machine
- All containers on a host share the host's kernel



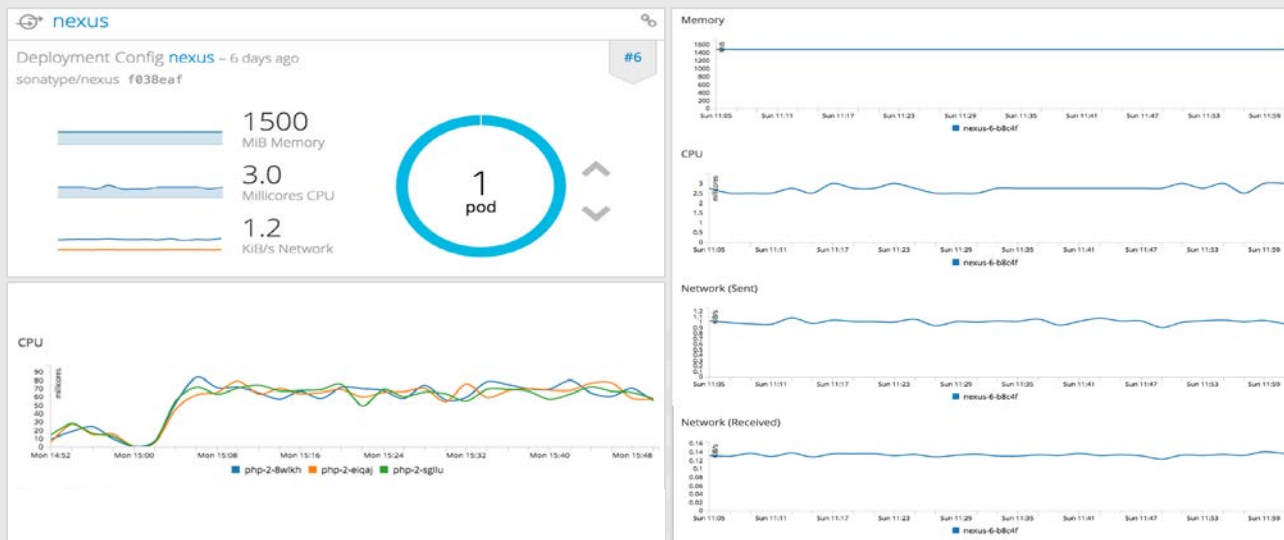


What's SELinux?

- A kernel security module for mandatory access control (whitelisting)
 - If something isn't explicitly stated as allowed, it's not allowed
 - Released in 2000 by the NSA
- Allows for dynamic isolation with MCS (multi-category security) labels

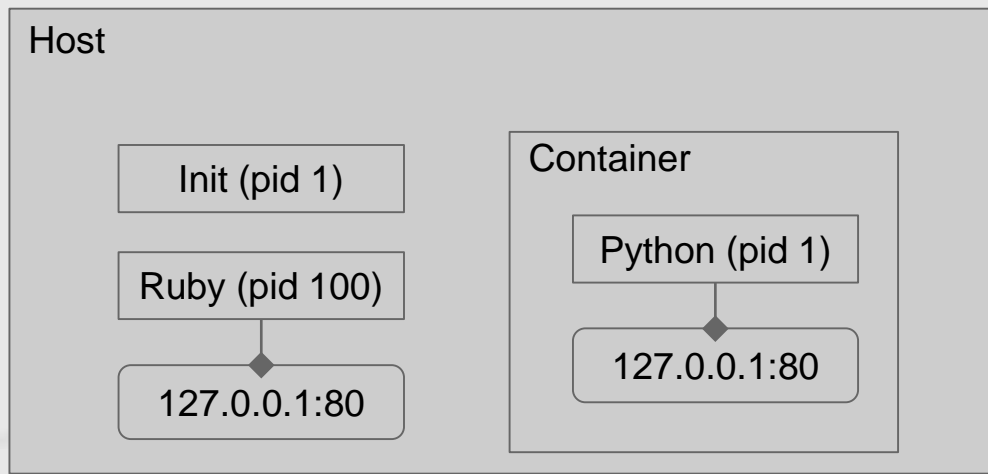
What's CGroups?

- Linux kernel feature to constrain resources (memory, cpu, network, disk) used by a process



What are Linux Namespaces?

- Linux kernel feature to isolate and virtualize system resources
 - mnt - Ex: /share
 - pid - Ex: ps -ef
 - net - Ex: 127.0.0.1
 - user - Ex: root



Why OpenShift/Kubernetes?

- Anything that runs in Linux can run in a Linux Container
- Makes your code/application:
 - Easy to distribute with lightweight immutable container image(s)
 - Isolated from dependency issues which makes your code more reusable
 - Easy to run with declarative config
 - Easy to operationalize with standardized infrastructure (logging, metrics, etc)
 - Provides complete application lifecycle management
 - Develop -> Build -> Test -> Run -> Manage
- Better utilization of hardware by reducing the number of operating systems and with overcommit scheduling and QoS tiers
- Container idling/auto-unidling



Benefits to Students/Researchers

- OpenShift/Kubernetes projects give isolation, quota, user management, and self service for dev->ops.
- Get into self service aspects that are more possible with OpenShift vs OpenStack
 - Imagine letting 10,000 technical students/teachers into Horizon
 - Will they understand the concepts?
 - Will they each get their own project?
 - If yes:
 - Who is going to manage their resource usage?
 - Who is going to manage security?
 - If no, will they play nicely with others and deal with shared quota?
- Security (projects, users, groups, roles, scc, linux capabilities, seccomp, selinux, routing tls, ...)
- Potential to get more resources because of overcommit



openstack®

Why OpenStack?

- OpenShift/Kubernetes runs everywhere so it standardizes deployments across all clouds (public and private)
- OpenShift and Kubernetes do not solve the infrastructure problem
- AWS/Azure/GCE/... all provide great public cloud services
- Containers need a unifying story for private cloud

Benefits of OpenShift on OpenStack

- Dynamic storage provisioning
- Readily available block storage (cinder) and object storage with Swift
- Shared auth through keystone
- Potential for cluster auto-scaling

...



OPENS SHIFT



openstack®

How could the integration be better?

- Removing doubly encapsulated networks (Kuryr)
- Better integration with storage
 - Snapshots
 - Automated backups that understand containers
- Better install/upgrade integration
- Better support in OpenStack for understanding containers in the UI
 - Understanding control plane vs nodes
 - Visualizing container application topology
- Linking between UIs depending on use case
 - OpenStack -> OpenShift/Kubernetes -> FaaS/Serverless

Thanks!