Understanding how Red Hat Enterprise Linux 8 provides a foundation for building and deploying containers

Dan Walsh
Senior Distinguished Engineer
May 2019
Please Stand
Please read out loud all text in RED
I Promise
To say
Make a copy
Rather than
Make a Xerox
I Promise
To say Tissue
Rather than Kleenex
I Promise
To say **Container Registries** rather than **Docker registries**
I Promise
To say Container Images Rather than Docker images
I Promise
To say Containers
Or OCI Containers
Rather than Docker Containers
Sit Down
THE JOURNEY

Single node

TRADITIONAL DEVELOPMENT

Find
Run
Build
THE JOURNEY

Single node+

TRADITIONAL DEVELOPMENT

Find | Run | Build

Share

#redhat #rhsummit
THE JOURNEY

Find
Run
Build
Share
Integrate
Deploy

TRADITIONAL DEVELOPMENT

CLOUD-NATIVE

Multinode
THE JOURNEY
Can start anywhere

TRADITIONAL DEVELOPMENT
Find  Run  Build

CLOUD-NATIVE
Integrate  Deploy

Red Hat® Enterprise Linux®
(Podman, Buildah, and Skopeo, Universal Base Image)

Red Hat Quay

Red Hat OpenShift® Container Platform (Kubernetes)
CUSTOMER NEEDS

CAPABILITY
- Single node
- Multinode

TECHNOLOGY
- Linux and container tools
- Linux and Kubernetes

PRODUCTS
- Red Hat Enterprise Linux
- Red Hat OpenShift
UNDERSTANDING CONTAINER TOOLS
Containers are processes—they run on a container host. **Containers are Linux.**

Containers are defined by the Open Containers Initiative (OCI), a collaborative project hosted by the Linux Foundation.

The docker daemon is one of the many user space tools/libraries that talk to the kernel to set up containers.
CONTAINER HOST PROCESSES

Regular processes, daemons, and containers all run side by side

Tightly coupled communication through the kernel—all-or-nothing feature support:

- Operating system (kernel)
- Container runtime (runC)
- Container engine
- The whole stack is responsible for containers—the container host
Container engines and runtimes rely on the kernel for storage:

- Cached container images map to layered file system
- Running containers often use an extra copy on write layer
INTRODUCING RED HAT ENTERPRISE LINUX 8 CONTAINER TOOLS MODULE
Each module defines its own life cycle, which is closer to the natural life of the application rather than the Red Hat Enterprise Linux life cycle.
THE CONTAINER TOOLS MODULE

One module delivered with multiple application streams based on different use cases:

- The Red Hat Enterprise Linux 8 stream delivers new versions for developers
- The versioned, stable streams provide stability for operations

![Graph showing updates by year for RHEL 8 fast stream, V1 and V2 stable streams.]
MAJOR TOOLS IN THE MODULE

- docker
- podman
- skopeo
- buildah
Replacing Docker With Podman

By Dan Walsh @rhatdan
dnf install -y podman
dnf install -y podman
alias docker=podman
Questions

Blog: https://podman.io/blogs

Github:
- https://github.com/projectatomic/libpod (podman)
- https://github.com/containers/storage
- https://github.com/containers/image
- https://github.com/containers/buildah

IRC: freenode: #podman
Site: https://cri-o.io
Alan Moran on Twitter: "I completely forgot that ~2 months ago I set up "alias docker='podman'" and it has been a dream. #nobufgataemons @projectatomic"

11:49 PM - 29 May 2018

7 Retweets 15 Likes
MODERN CONTAINER TOOLS
Providing stability, flexibility, and performance with containers and images

Container-tools – OCI tooling to create, run, and manage Linux containers with an enterprise life cycle

- Conform to the OCI image and runtime specifications
- Daemon-less, OS-native container tooling
- Separation of concerns
CONTAINER TOOLS
Improved security model, stability, and life cycle

podman

A daemon-less CLI/API for running, managing, and debugging OCI containers and pods

- Fast and lightweight
- Uses runC
- Provides a “docker-style” syntax for working with containers
- Standard CNI networking
- Remote management API via Varlink
- Provides systemd integration and advanced namespace isolation

RED HAT ENTERPRISE LINUX KERNEL

#redhat #rhsummit
CONTAINER TOOLS
Improved security model, stability, and life cycle

buildah

A daemon-less tool for building and modifying OCI/Docker images

- Preserves existing Dockerfile workflow
- Allows fine-grained control over image layers, the content, and commits
- Lets you minimize container images by using tools from the container host rather than adding them in the image
- Shares the underlying image and storage components with Podman and CRI-O

Start from an existing image or from scratch

Generate new layers and/or run commands on existing layers

Commit storage and generate the image manifest

Deliver image to a local store or remote OCI/Docker registry

#redhat #rhsummit
CONTAINER TOOLS
Improved security model, stability, and life cycle

**skopeo**
A comprehensive tool and library to inspect, sign, and transfer images.

- Inspect image manifests
- Sign and verify image manifests
- Push/pull images
- Currently the only tool capable of copying images between registries
-Disconnected environments
- Same code base as the/containers/image library which is used by buildah, podman, and CRI-O
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat