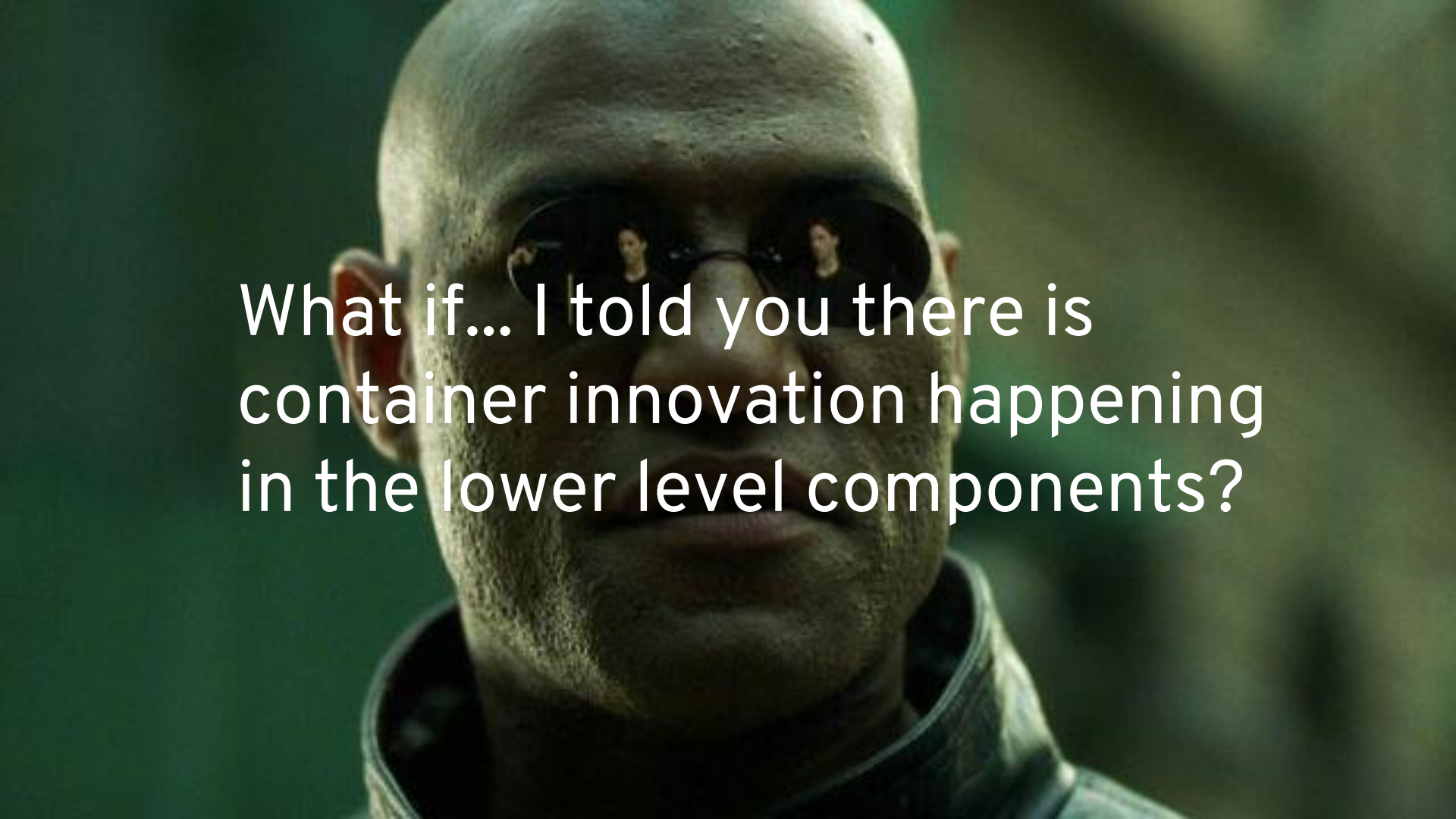




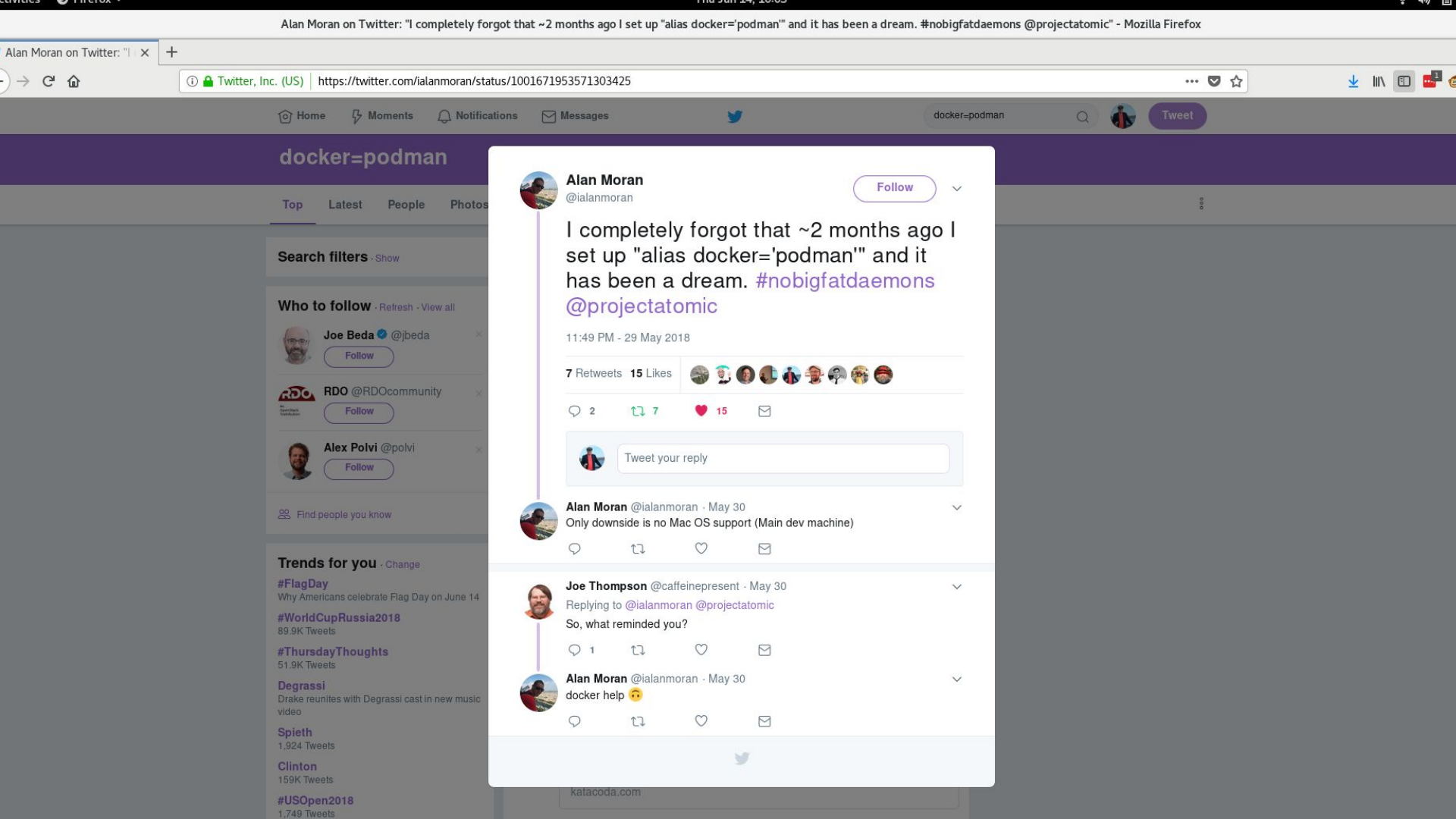
THE STATE OF CONTAINERS

Engines & Runtimes in RHEL & OpenShift

Scott McCarty
Principal Technology Product Manager - Containers
10/15/2018



What if... I told you there is
container innovation happening
in the lower level components?



docker=podman

Top Latest People Photos

Search filters · Show

Who to follow · Refresh · View all



Joe Beda · @jbada

Follow



RDO · @RDOcommunity

Follow



Alex Polvi · @polvi

Follow

Find people you know

Trends for you · Change

#FlagDay

Why Americans celebrate Flag Day on June 14

#WorldCupRussia2018

89.9K Tweets

#ThursdayThoughts

51.9K Tweets

Degrassi

Drake reunites with Degrassi cast in new music video

Spieth

1,924 Tweets

Clinton

159K Tweets

#USOpen2018

1,749 Tweets



Alan Moran

@ialanmoran

Follow

I completely forgot that ~2 months ago I set up "alias docker='podman'" and it has been a dream. #nobigfatdaemons @projectatomic

11:49 PM · 29 May 2018

7 Retweets 15 Likes



2



7



15



Tweet your reply



Alan Moran · @ialanmoran · May 30

Only downside is no Mac OS support (Main dev machine)



Joe Thompson · @caffeinepresent · May 30

Replying to @ialanmoran @projectatomic

So, what reminded you?



1



Alan Moran · @ialanmoran · May 30

docker help 🙄



What Kinds of Things, You Ask?

- Small core-utils style approach
- Running without root
- Easily move between Podman and Kubernetes
- Build Images with Declarative Languages (Ex. Ansible)
- Enabling security compliance
- Virtual machine isolation
- Multiple architectures
- Application Specific Security Profiles

WHY CONTAINERS

Historic Challenges



- IT Delivers many different applications
- Each requires different languages, and libraries
- Deploy, configure, manage and maintain is complex
- Expensive - time and money

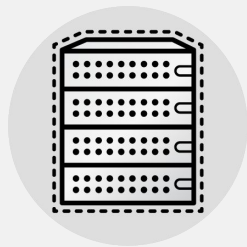
Why Containers



- Each application comes with its own dependencies
- Portable and consistent application environments
- Developer choices don't interfere with host
- Operations choices don't interfere with applications

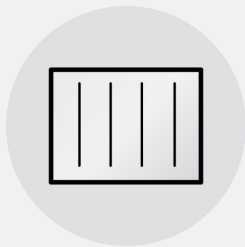
We already have the technology...

Not exactly, it's about finding, running, building, sharing, integrating, and deploying services



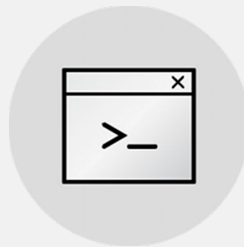
Virtual Machines

Good at exposing resources (CPU, RAM) but collaboration is like emailing Word docs



Containers

Just enough. Easy to collaborate, rebuild, combine and share with others



PaaS

Good at quick deployments, but too inflexible to modify services easily

The Journey

Single Node



Traditional Development

FIND

RUN

BUILD

The Journey

Single Node+



Traditional Development

FIND

RUN

BUILD

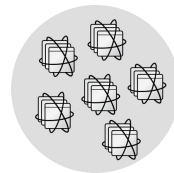
SHARE

The Journey

Multi Node



Traditional Development



Cloud Native

FIND

RUN

BUILD

SHARE

INTEGRATE

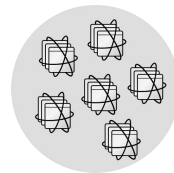
DEPLOY

The Journey

Can start anywhere



Traditional Development



Cloud Native

FIND

RUN

BUILD

SHARE

INTEGRATE

DEPLOY

Podman/Buildah/Skopeo

Quay

OpenShift

WHAT RED HAT PROVIDES

Customer Needs

Mapping customer needs to solutions

Capability

Single Node

Multi Node

Technology

Linux &
Container Tools

Linux &
Kubernetes

Product

Red Hat
Enterprise Linux

OpenShift

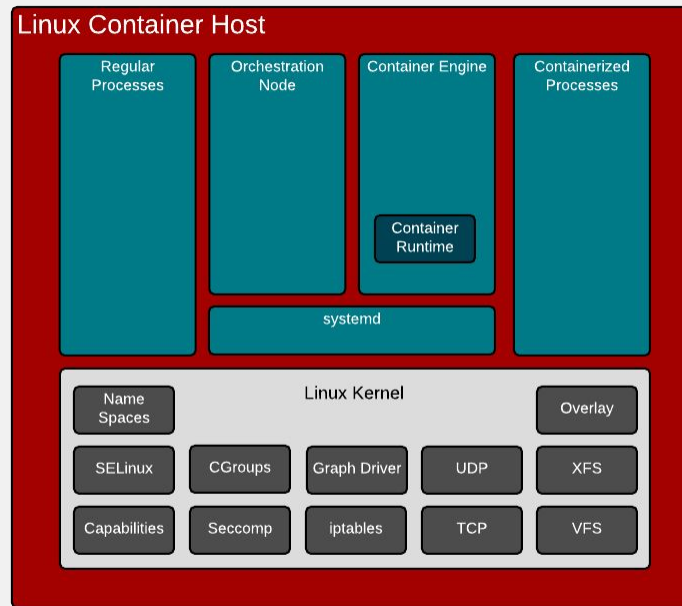
What are the lower level components?

The foundation for OpenShift

Tightly coupled communication through the stack

- all or nothing feature support:

- Orchestration Node (Kublet, OpenShift Node)
- **Container Engine** (Docker, Podman, CRI-O)
- **Container Runtime** (runc, gVisor, Kata)
- Operating System (Linux kernel)



Container Engines

Pluggable within most container orchestration



cri-o



docker



rkt



buildah



podman

Why & What:

- Run & Build containers
- Many, many, projects...
- What are Moby, Docker CE, and Docker EE?
- CRI-O gaining popularity
- Podman, Buildah, Skope are exciting

Container Runtimes

Pluggable in within most container engines

 opencontainers / **runc**

State of container runtimes:

- 99% of the world uses *runc*
- Pluggable because of the OCI Container Runtime Specification
- Kata & gVisor both look interesting
 - Red Hat is engaged upstream
 - Not mature enough to be on our product roadmaps

 **kata**containers

gVisor

OPEN CONTAINERS

Containers Are Open



Established in June 2015 by Docker and other leaders in the container industry, the OCI currently contains three specifications which govern, building, running, and moving containers.

Standards Are Well Governed



- Governed by The Linux Foundation
- Ecosystem includes:
 - Vendors
 - Cloud Providers
 - Open Source Projects

Overview of Important Standards

Vendor, Community, and Standards Body driven



Open Containers Initiative (OCI)
Image Specification



Open Containers Initiative (OCI)
Distribution Specification



Open Containers Initiative (OCI)
Runtime Specification



kubernetes

Container Runtime Interface
(CRI)



CNI

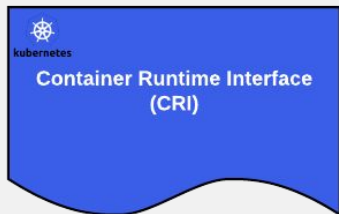
Container Network Interface
(CNI)

Many different standards

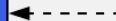
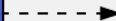
CRI-O & CRICTL

Interfaces for humans and robots

Governing
Standard

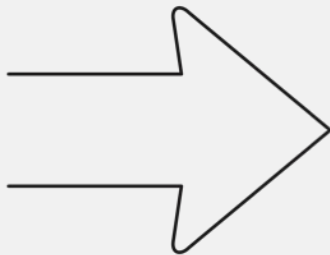


Workflow



RED HAT INVESTMENT

Container Engine/Runtime Strategy





cri-o

Experience:

- A lightweight, OCI-compliant container runtime designed for Kubernetes
- Runs any OCI compliant, Docker compatible container images
- Focus on stability and life cycle *with* the platform
- Improve container security & performance at scale

Roadmap

- Now [running in production](#) under OpenShift Online clusters
- Permanent Kubernetes project
- Continues to track and release with upstream Kubernetes
- On track to become the default container engine for nodes
- Converting node troubleshooting documentation to use crictl for human interface to CRI-O
- Adding user namespace support
- Integrating libpod for better CLI integration with Podman



buildah

Experience

- Will be embedded in OpenShift build strategies, mostly transparent (except custom build strategy)
- OCI Container images compatible with Docker format
- Multi-stage builds supported with and without dockerfiles
- Customizable image layer caching
- Shares the underlying image and storage components with CRI-O

Roadmap :

- GA support with RHEL 7.5
- User namespace enablement
- Working towards unprivileged, non-root container builds
- Future integrations with Ansible (new work on Ansible Builder), and OSBS



podman

Experience

- Provides a familiar command line experience compatible with the docker cli
- Great for running, building, and sharing containers outside of OpenShift
- Can be wired into existing infrastructure where the docker daemon/cli are used today
- Simple command line interface, no client-server architecture, so more agile in many use cases

Roadmap:

- GA in RHEL 7.6
- Run containers as non-root (enhanced user namespaces)
- Docker compatible health checks
- Atomic run label support

CLOSING

Please Stand

Please read
out loud all
text in
RED

I Promise

To say
Container Registries
Rather than
Docker registries

I Promise

To say
Container Images
Rather than
Docker images

I Promise

To say
OCI Containers
Rather than
Docker Containers

Please Sit

Q & A

Source Material

Presentations, Blogs, Etc

List:

- [State of Container Technologies in the Operating System](#) - Dan Walsh 10/2018
- [Collabzone - Red Hat Container Engines, Tools, and Images](#) - Scott McCarty 10/2108
- [High Touch Beta \(HTB\) for RHEL 7.6: What to expect with podman?](#) - Dan Walsh 09/2018
- [RHTE - Docker What? Buildah, Podman et al](#) - William Henry 09/2018
- [Red Hat Container Tools](#) - Ben Breard & Scott McCarty 10/2018



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHat



youtube.com/user/RedHatVideos