OpenShift 4.12 - It’s a Linux system again

“CoreOS Layering”/ostree-native containers

Colin Walters

A brief history of time

We’re on a journey - with you

The OpenShift 3 to OpenShift 4 was a big leap after the acquisition of CoreOS. We didn’t know if it would work.

- Opinionated installer
- Cluster version operator, release image, second level operators
- RHEL CoreOS - ignition, etc. (cross cloud and bare metal!)
- Machine Config Operator

Configuring the OS via Kubernetes

Our first take

You can configure/change your kernel arguments via oc/kubectl! (kernelArguments in MachineConfig) - We roll out that change to the targeted nodes, handling draining etc. 🎉

But…

It’s a ~~Unix~~Kubernetes system!

This was **not** a scene in Jurassic Park (1993, Linux 1991)

“Linux system” vs Kubernetes

ssh, systemd, single node vs kubectl/cluster

- Should you enable ssh or not?
- Even in a cluster, we have to accommodate machine-specific state like static IPs
- Installing not-containerized agents/monitoring/storage
- OpenShift SDN team pushed hard to go back to shipping openvswitch on host via systemd
- Actually, even the giant leap between docker/podman vs kubernetes is still a real problem

Spanning the gap: build OS from base image

We're making {Fedora,CentOS,RHEL} CoreOS a **native container base image**

```
FROM quay.io/…/rhel-coreos-8

RUN rpm -Uvh
http://mirror.centos.org/centos/…/iptables-1.8.4-23.el8.x86_
64.rpm && ostree container commit
```

Push to a registry - **then you boot it.** There's a kernel in there and some ostree stuff.

(You can run it as a container in any OCI runtime too, to e.g. inspect things of course)

Boot it…how?

OK so we put ostree and the container stack in a blender

- Reworked (rpm-)ostree stack to learn how to fetch container images via skopeo **directly**
- But bootable state is still in /, distinct from /var/lib/containers
- Keeping all the ostree stuff “behind the scenes” that we still need, like kernel and kargs management
- Enables a seamless transition - in fact, this will be a mandatory and (hopefully) entirely invisible change by default for RHCOS upgrades in 4.12

I made a container, now what?

Telling the MCO to roll out your custom OS image is this easy

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 95-workers-hotfix
spec:
  osImageURL: quay.io/examplecorp/rhel-coreos-8-custom@sha256:5891…
```

You’ve got questions

The docs will have answers

- Cluster upgrades become decoupled from OS upgrades
- But…removing your override image resets reliably back to “golden” base image
- Applying other MachineConfig on top continues to work
- A lot more to come

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat