Unsung Hero of the Cloud Native Revolution
Container Linux Then and Now

KubeCon NA 2022

Vincent Batts

Howdy

```
$> whoami
vbatts
$> id -Gn
devel msft flatcar kinvolk redhat oci appc docker
golang slackware ruby
```

Container Linux!

- Cloud Native!
- Minimal Footprint!
- IKEA vs. Artisanal
- Atomic updates!

But, what is “a Linux”?

- Well, actually it’s GNU/Linux …
- What makes it “what people and tooling expect?”

Is this a Linux?

- syscalls, ioctls, fcntls?
- libc?
- Kernel CONFIG_’s?
- Config (/etc or CM tooling)
- Early boot provisioning? (ignition or cloud-init) ((secrets??))
- Host services? (unit files and dbus?)
- NVR of software

A brief history

- Read only?
  - Knoppix (2000) – a first of the “live” from CD Linuxes
- Minimal?
  - Damn Small Linux (2005) ~50Mb
- Isolation?
  - mountns (2002)
- “Userspace”
  - An ongoing swarm of iteration and improvements and sprawl

A brief history

- 2008 LXC is launched
- 2013 CoreOS releases the first “Container Linux”
  - based on the ChromiumOS created for Chromebooks
  - Gentoo is the upstream
- 2013 dotCloud releases Docker
- 2014 Kubernetes is launched
- Since then: Atomic; Flatcar; Talos; FedoraCoreOS (RHEL and CentOS too); bottlerocket; k3os; openSUSE Micro; LinuxKit; gardenlinux; and I’m sure others ...

(other history talks at github.com/vbatts/talks)

Common Challenges

- Do you know when something becomes *artisanal*?
- Package management
  - Trust
  - Proofs
  - Determinism (scriptlets; ordering; file system API's)
- Who says a a reboot is needed? How?
- "Let's innovate and think about how to migrate later!"
- "If folks should use **my** approach if they want things to work together!"

Common Challenges

- “is this change/rollout deterministic?”
- When you deploy “A Kubernetes”, that is not a definitive, standard *thing*.
  - Every extension, runtime, plugin, and even config field can have implicit and explicit variations needed.

Don’t get me wrong

- Packagers and Upstreams are **critical**
- The kernel and OS [interfaces] you use **matter**
- You end users and admins/ops/architects *actively* don’t care about your epiphany if you’ve made their life harder for no reason
- Everyone wants ***their precious*** to be the winner …

Where are we now?

- It depends...
- Compartmentalizing our frustration
- We’ve all suffered these pains, individually and collectively
- Renewed interest in defining and establishing the common OS interfaces
  - #kube-operating-systems-dev (k8s slack)
  - (beyond kubelet, CRI, CNI, CSI, etc.)
- What is your use-case? Let that drive you, not a marketing pitch

y’all

- Find me online
  - Then we can talk like humans
    - Let’s work together