Building Fedora CoreOS

Presented by
Benjamin Gilbert, Dusty Mabe
Red Hat

Container Linux Goals

History

- Based on ChromiumOS
  - Inherited many of its design choices
    - including Gentoo upstream
- Epoch: July 1, 2013
- 85 stable releases, 120 beta, 283+ alpha

Goals

- Minimal container-focused server OS for production use
- Clustered deployments
- Immutable infrastructure
- Automatic updates
- Broad platform support
- Stable and secure

Production OS

- No in-image development tools
  - ...except Git
- Not a lot of in-image debugging tools, either
- Just enough to talk to your hardware and run containers

Immutable infra

- Node customizations are all encoded in the provisioning config
- You can modify the node afterward, but don't
- Configuration management is an anti-pattern

Automatic updates

- Users shouldn’t have to think about updates
- No backward compatibility breaks, ever
  - Old config files must work with new binaries
  - Must support old bootloaders, cloud agents
  - Long deprecation windows for services
- No regressions, ever
  - ...or users will disable updates
  - This is the hard part

fedora

Platform support

- Bare metal: install to disk, live PXE
- Clouds: AWS, Azure, DigitalOcean, GCE, Packet, others
- Virt: QEMU, VirtualBox, VMware, OpenStack, Xen

Container Linux Install and Update

Partition table

- /boot (ESP)
- /usr (USR-A)
  - Immutable OS image (protected with dm-verity)
- /usr (USR-B)
- /usr/share/oem (OEM)
  - Platform-specific customizations
- / (ROOT)
  - User data, including /etc

Install process

- The cloud doesn't have installers
- Bare metal shouldn't either
- coreos-install: 350 lines of shell
  - Runs on any distro
  - Download, decompress, verify, dd image to disk
  - Doesn't work on 4Kn drives

Provisioning

- Still need to do the things an installer would do
- coreos-cloudinit
  - That Other cloud-init is written in Python
  - Write our own! sorta compatibly
  - Reconfiguring the system halfway through boot is a bad idea
  - Especially when it fails
  - Runs on every boot
    - Users try to use it for configuration management

Better Provisioning

- Ignition
  - Runs in the initramfs, on first boot
  - Fetches userdata from the usual places
  - Can drop files, systemd units, create users & groups, create partitions and RAID volumes, reformat your root filesystem
  - If provisioning fails, so does boot

Atomic updates

- Active+passive /usr partitions
  - Update payload is a kernel and an ext4 filesystem
  - Easy to understand, but inflexible
    - Updates overwrite previous version
    - Rollback is possible for 45 seconds after boot
- One update image per architecture
- Many install images with different OEM partitions

Update system

- Staged rollouts
  - Allows monitoring of update status
  - but really, gives users time to report bugs
- Some useful metrics
  - Breakdown by cloud platform and version
  - Also stores original install version, last checkin, update status
  - If users turn off updates, we don't get metrics

Update system

- Omaha protocol
  - ...modified
- update_engine
  - Complex and unmaintained
- CoreUpdate server
  - has scaling problems

Updating a cluster

- Cluster assumed redundant
- Okay to reboot nodes on upgrades
  - So long as we don't reboot them all at once
  - locksmith: reboot coordination via etcd
- Update system doesn't talk to the cluster
  - Nodes don't drain before reboot
  - Cluster doesn't control OS version
- Need cluster-level coordination
  - ...which doesn't get any help from the OS

Automatic updates

- Except the OEM partition
- and the bootloader
  - Stray memcpy in GRUB

Automatic rollback

- ...kinda.
- Kernel panic on boot → roll back
- Fail to mount root FS → roll back
- Fail to start network → broken machine
- Fail to start important service → broken machine
- Roll back → try to upgrade again
  - and don't report the failure anywhere
- No user-specified health checks

fedora™

Update channels

- Stable
  - Security updates, bug fixes
- Beta
  - Staging for stable
  - Security updates, bug fixes
- Alpha
  - Staging for beta + bleeding edge software
  - Security updates and bug fixes
- Regular promotions plus out-of-cycle updates
- All channels are expected to work!

fedora

Testing

- CI only; no routine manual testing
- Users should run alpha and beta in their environments

Container Linux Runtime

CPU architectures

- ARM64 added later
- Never reached feature parity with AMD64
- Bolting on multi-arch support doesn’t work (!)

Container-optimized

- Do not run things in the host
- Do not run things in the host
- We will break things that run in the host
- ...out-of-tree kernel modules?

Container engines

- Everyone wants the latest version of Docker
- ...except Kubernetes
- torcx: Roughly half a package manager

No interpreters

- Except bash
  - and awk
- Good
  - Don’t run stuff in the host!
  - Smaller image
  - Smaller attack surface
- Bad
  - Have to rewrite convenient tools in inconvenient languages

Platform agents

- Carried in the OEM partition
  - ...sometimes including Python
- Cannot be updated
- Sometimes unpleasant

Atomic Host Design Goals

Update Model

- Download update in the background
- Stage new deployment for next reboot
- Boot into upgraded deployment

Design Goals

- Reliable (fault tolerant) Updates
- Offline Updates
- Security

Design Goals

- Reliable (fault tolerant) Updates
- Offline Updates
- Security

How?

fedora

Design Goals

How?

- Shrink the base
- Leverage containers
- Develop image based update system

fedora

Design Goals

- *Reliable (fault tolerant) Updates*
- *Offline Updates*
- *Security*
- **Good Container Host**

Design Goals

- Reliable (fault tolerant) Updates
  - Update Model Allows for this
    - *Download update in the background*
    - *Stage new deployment for next reboot*
    - *Boot into upgraded deployment*
  - Easily roll-back if new update doesn't work
    - Userspace: `rpm-ostree rollback`
    - Boot-loader: 2nd boot-loader entry for fallback

Design Goals

- Offline Updates
  - Update Model Allows for this
    - *Download update in the background*
    - *Stage new deployment for next reboot*
    - *Boot into upgraded deployment*
  - No software ever runs in half upgraded state
  - No need to worry about older versions of software (CVEs) in running apps (in memory)

Design Goals

- Security
  - Smaller base == Less risk
  - Image based update system
    - Able to verify server side content matches local checkout
  - Mount filesystems read only
  - Leverage SELinux

Design Goals

- Good Container Host
  - Provide Container Runtime(s) for users
  - Host updates managed by Atomic Host team
  - Application updates managed by Admin
    - Applications run in containers
  - Separation of concerns increases reliability of host updates

Atomic Host Structure

- Content Tracking (RPM-OSTree)
  - Hybrid Approach (not a disk image)
    - Sits above filesystem
    - Knows about the software contents (rpms)
    - Knows about bootloaders
  - “git for your OS”
    - ostree repo is like a git repo
      - refs/branches to follow
      - checkouts
      - Rebasing
    - Share common contents between different deployments
      - “deduplication”

fedora

Atomic Host Structure

- Disk Layout
  - Generic Disk Layout approach
    - Allows user to configure storage for system
      - Partition Based
      - LVM
      - BTRFS
    - Configured during install
  - Mount points
    - /usr **READONLY**, /var **READ/WRITE**
    - {/home,/mnt,} → /var{/home,/mnt,}
    - State in /etc is tracked and restored on rollback

fedora

Atomic Host Structure

- Bootstrapping
  - Bare Metal
    - Use Installer ISO (anaconda based)
  - Cloud
    - Use cloud-init (baked into OSTree)

Atomic Host Structure

- Host Extensibility
  - Package Layering
    - Add an RPM to the OSTree as a **layer**
    - Mutates the immutable Host in a controlled way
  - System Containers via Atomic CLI
    - Grab container images from OCI registry
    - Easily set up systemd units to run them on boot
    - Often super privileged (lots of hooks into the Host)

Atomic Host
What Worked Well

Atomic Host: Good

- Using Fedora/rpm ecosystem
  - Consume RPMs, participate in Fedora
- Package Layering
  - Containerizing low level system tools is hard
  - Package layering helps us get around these issues
- “Git for your OS” model
  - Easy to understand conceptually

Atomic Host: Good

- Representing system state in a clear way

```
[dustymabe@dhcp137-98 logs]$ rpm-ostree status
State: idle; auto updates enabled (check; last run 15h ago)
Deployments:
● ostree://unifiedrepo:fedora/28/x86_64/atomic-host
                   Version: 28.20180722.0 (2018-07-23 00:38:05)
                BaseCommit: 106a7095fd46741948ba60dbd0586668ae7abe6336671444ec790d7541a88778
              GPGSignature: Valid signature by 128CF232A9371991C8A65695E08E7E629DB62FB1
           LayeredPackages: aria2 git git-annex mosh pciutils tig vim
```

fedora

Atomic Host What Didn't Work Well

From Me <dusty@dustymabe.com>

Subject **Top Issues facing our Atomic Host users**

To atomic-devel@projectatomic.io

Bcc Me <dustymabe@gmail.com>

From the past few months we've had a lot of community involvement in IRC and on the mailing lists. Over that time there are some re-occurring issues that keep coming up.

- Lost content in */etc/* problem
    - https://github.com/projectatomic/rpm-ostree/issues/40
    - https://github.com/ostreedev/ostree/issues/545
    - TL;DR - to avoid this problem don't make changes to */etc/* after you've staged a new deployment. i.e. you should reboot soon after you've `rpm-ostree upgrade` or `rpm-ostree install`
    - We are hoping to fix this soon!

- Can't package layer rpms that store content in */opt/* or /usr/local
    - https://github.com/projectatomic/rpm-ostree/issues/233
    - TL;DR - you can't package layer rpms that own content in /opt or /usr/local
    - We are hoping to fix this but the best approach has not yet been determined. Join in the upstream issue to discuss

- Out of Tree Kernel modules; support for DKMS/akmods
    - https://github.com/projectatomic/rpm-ostree/issues/1091
    - TL;DR - if you have out of tree kernel modules and want an 'autorebuild' system like DKMS or akmods to work you are currently out of luck.
    - This one is currently not high on the radar. It's going to be a ton of work to get right.

- split base/layered versionlocked package problem
    - https://github.com/projectatomic/rpm-ostree/issues/415
    - TL;DR - This happens when your base layer is out of date with the rpms from the yum repo it is pulling the new rpms from. Most of the time you can fix this by running `rpm-ostree upgrade` and then `rpm-ostree install`. There are some cases where we have not yet done a new release yet and there are packages that would be overridden. For this we are discussing in the upstream issue to allow a --force flag. Please add input to the upstream issue.

Atomic Host: Bad

- Lost content in /etc/ problem
  - Changes to /etc after `rpm-ostree upgrade` lost
  - Fixed now with staged deployments
- Can't package layer some RPMs
  - RPM-OSTree strict about contents (/opt, /usr/local)
- 3rd Party kernel modules
  - DKMS/Akmods – Don't work

fedora

Atomic Host: Bad

- Automatic Update Philosophy
- Package layering makes upgrades less reliable
  - We can only reliably test upgrades of base content

Fedora CoreOS

Goal

"An automatically updating, minimal, monolithic, container-focused operating system, designed for clusters but also operable standalone, optimized for Kubernetes but also great without it."

fedora™

Use cases

- Primary
  - Clustered server node for running Kubernetes/OKD
  - Single server node for running containerized applications
- Secondary
  - Clustered server node for running non-Kubernetes container orchestration platforms

Platforms

- Same primary platforms as Container Linux

Install process

- Something like coreos-install
- Fetch image and write to disk

Provisioning

- Ignition

Partition layout

- Maybe one root partition, maybe root + /var

Automatic updates

- rpm-ostree, Fedora RPMs
- Rate-limited rollouts
  - Not with CoreUpdate or Omaha
- Cannot break users, ever
- Automatic rollback with user-provided health checks

fedora

Update streams

- Details TBD
- There will be pre-stable streams

Testing

- CI
- User feedback from pre-stable streams

Metrics

- We will have some
  - Helpful for directing development effort
- New system, since we’re not using CoreUpdate
- There will be privacy knobs

Container infra

- Runtimes
  - Still need a way to ship both current and ancient Docker
  - ~~rkt~~
  - Podman
- Probably ship kubelet and CRI-O
  - But they care about the Kubernetes version

Package overlays

- Useful for debugging
- Maybe needed for alternate Docker/kubelet/CRI-O
- Discouraged for general use

ARM64

- Try to ship it

Python

- Try not to ship it

Cloud agents

- No separate OEM partition
  - One update image, multiple almost-identical install images
- Ship agents until we can replace them

More thought needed

- Better cluster coordination
  - Reboot coordination, cluster version management
- Third-party kernel modules

Thanks!