Modding the immutable
How to extend Flatcar, an immutable image-based OS

FOSDEM²² | 2022-02-06
Hi, I’m Thilo

Thilo Fromm
Engineering manager, Microsoft

Github: t-lo
Twitter: ThiloFM
Email: thilofromm@microsoft.com
Agenda

Brief Intro to Flatcar Container Linux

Build your own OS image

Make modifications build your own custom image

Brief note on testing
Flatcar Container Linux primer

“Flatcar Container Linux is a fully open source, minimal-footprint, secure by default and always up-to-date Linux distribution for running containers at scale.”

Minimal, immutable base system
Fixed set of tools, no add-ons
Strong focus on container workloads
Strict separation between OS and Apps
Flatcar Container Linux primer

Ships as image - no package management

OS part of image is immutable – no run-time mods

Updates ship as images only, too
Flatcar Container Linux primer

Excursion: partitions after provisioning (simplified)
Flatcar Container Linux primer

All OS binaries & libraries reside in /usr

/usr is read-only

OS is treated like a single application
   (also see containers devroom talk on automated updates)

Regular, rapid releases
   (new major Alpha: 2 weeks, Beta: 1 month, Stable 2-3 months)
Modifying the OS

We will vend our own OS image, using the SDK

Flatcar is based on (...) Gentoo, i.e. built from sources

However, we provide bin pkg servers for releases
Modifying the OS – other options

...which we won't look into today 😞

changing the OS at runtime (e.g. adding binaries)

adding custom kernel modules
via writable overlay and the dev-container
Modifying the OS - goals

Add or update packages and file upstream PRs
  (additions must adhere to the Flatcar mission statement)

Fix bugs you've hit or help fixing bugs others ran into

First step of becoming your own upstream!
Modifying the OS

„scripts“ repo: entrypoint for Flatcar development

Wrapper script for using the SDK container

Git submodules w/ ebuilds (pkg build descriptions)
Modifying the OS

Pick a version
Modifying the OS

Pick a version
Modifying the OS

Pick a version
Modifying the OS

Pick a version
Modifying the OS

Pick a version
Modifying the OS

Pick a version
Modifying the OS

Pick a version – recommended: the latest Alpha release

Solid foundation: Image builds and tests OK

Binary packages available

Reasonably close to main („alpha-next“)
Modifying the OS

Build the stock OS image to validate all's well

./build_packages
./build_image

(go fetch a coffee – packages take ~1h, image ~30min)

./image_to_vm.sh
Let’s add a package

Gentoo upstream ships a vast variety of packages

We’ll tap that source; minor ebuild skills required

Live the dream – and ship screen!
Modifying the OS

Where are the packages?

- Scripts
- Git repo
- Build scripts + packages + image
- Rebuilding the SDK
- SDK-container/3rd-party/src
  - CoreOS-overlay
  - Portage-stable

Git-submodules

Git-submodules
Modifying the OS

Where are the packages?

/home/sdlc/trunk/src/scripts/

SCRIPTS Git Repo ← BIND-MOUNT FROM HOST

Build Scripts & Packages + IMAGE

RE-BUILDING THE SDK

SDK-CONTAINER/3RD-PARTY/SRC

PACKAGE BUILD

INSTRUCTIONS (ebuilds)

COREOS-OVERLAY

PORTAGE-STABLE

Git-Submodule

Git-Submodule
Modifying the OS

We’ve just created our very own personalised OS image

We introduced a new package based on upstream ebuild

We booted the OS and validated the changes
A brief note on testing

Flatcar SDK ships a thorough test suite – kola

Kola is complex, worth of its own talk
(see the talk „All's well that tests well“ by Mathieu Tortuyaux)

```
sudo kola run --platform=qemu \
  --qemu-bios=/usr/share/qemu/bios-256k.bin \
  --qemu-image=flatcar_production_image.bin \
  --tapfile=test-result.tap \
  cl.basic
```
Thank you 🙏

This talk as a doc:  

Contribute – https://github.com/flatcar-linux/Flatcar