facebook

cpe

Fedora @ Facebook

Michel Alexandre Salim

Client Platform Engineering

FAS: salimma / IRC: michel_slm

A pipeline to upstreams

Agenda

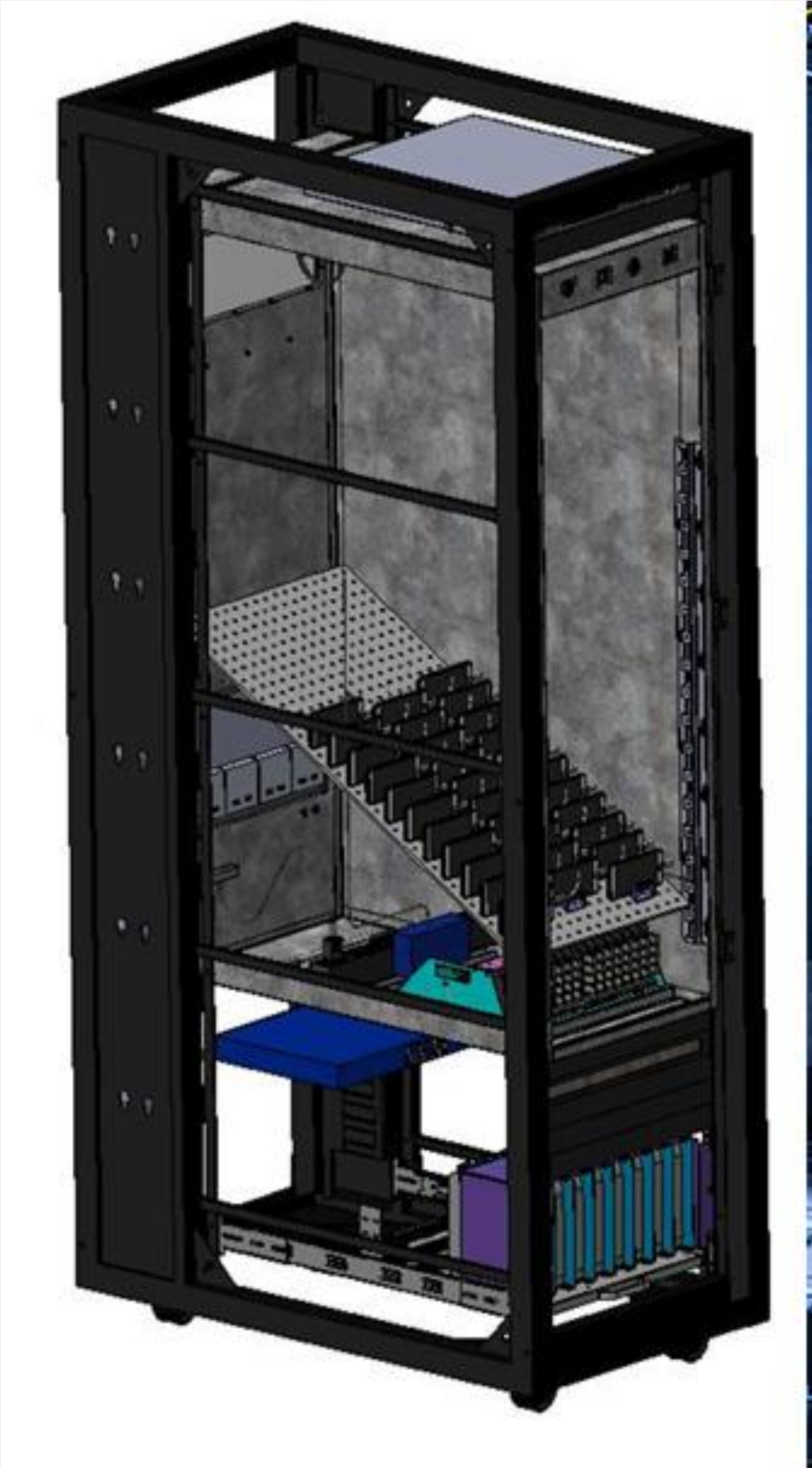
- Introduction
- Our two upstreams
- Provisioning revamp
- Dogfooding Fedora 33 Changes
- Upcoming projects

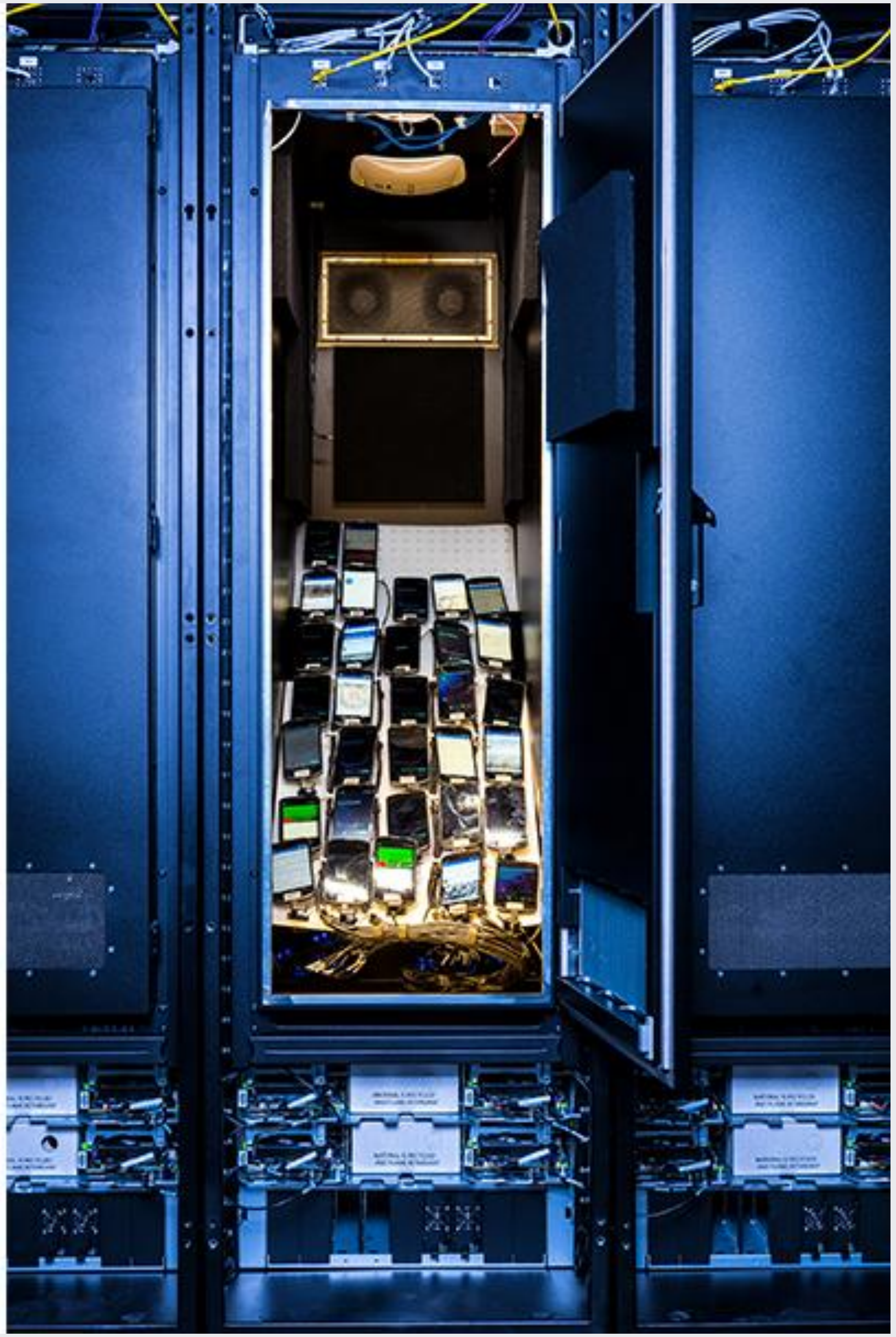
Intro

Who am I?

- Linux user since 1998
- Fedora contributor since 2005
- Facebook Production Engineer
- on my second team at Facebook

from managing a fleet of these...





to managing these...

Intro

Just kidding! (those were our cats)

- I am part of the Client Platform Engineering team
- Focusing mostly on managing the Linux client fleet
- am I a masochist? ¯_(ツ)_/¯

Facebook’s Desktop Linux fleet

- ~ thousands of Linux laptops and desktops (mostly Fedora)
- compare to ~ millions of Linux servers (CentOS)
- using similar distributions allow reuse
  - packaging pipelines
  - configuration management cookbooks
- mostly Lenovo (ThinkPads and ThinkStations)
- evaluating CentOS for desktop use

Our two upstreams

Upstream #1: Fedora

- we stick to Fedora defaults as much as possible
- provisioning using kickstarts
    - needed for consistency as we require full disk encryption
    - LUKS cannot be retrofitted
- we dogfood upcoming changes when they make sense

Upstream #2: the production fleet

- CentOS 7, ongoing migration to CentOS 8 Stream
- very slight modification e.g.
  - kernel (maintained internally)
  - systemd (to track upstream)
- see facebookincubator/rpm-backports
- resource control with cgroups2 (internally: fbtax2)
- fleet managed using Chef

collaboration

Collaborating with upstreams

- Fedora: test upcoming changes in a reasonably controlled environment
- prod: validate server changes on desktop environments
- EPEL
  - needed for some prod workflows
  - would be increasingly critical if we deploy CentOS on desktops

Organization

- organizing the individual Fedora contributors working at FB
  - e.g. a packaging oncall
- involving user groups that use non-standard hardware (e.g. MSI’s Killer® WiFi) with Fedora test days and reviewing updates in Bodhi
- working with Fedora and other FB teams on change proposals

time for some examples

provisioning: before

- PXE -> iPXE
    - no Secure Boot
- assumes internal network
- hard to test and deploy kickstart changes
- hard to produce repros for external bug reports
- **WFH**

provisioning: after

- modular
  - uses pykickstart’s ksflatten and ksvalidate
  - can build a kickstart without internal bits for repro
- WFH provisioning
  - we ship a script to bootstrap access to internal network
- kickstart injected to netinstall ISO
  - using lorax’s mkksiso - fixed a UEFI boot issue
- can test on VMs!

Demo Time

Things are changin’

Btrfs By Default: Before

- LUKS for full disk encryption
- used default Workstation layout (LVM, root, home, swap)
  - root tends to run out of space
- switched to unified root+home (LVM, root, swap)
- can’t reimage without backing up

Btrfs By Default: After

- still using LUKS
  - waiting for Btrfs native encryption to drop this
  - will actively encourage people to reimage once this is ready
- Btrfs + swap on ZRAM
  - might need to add swapfile
  - will have to rethink hibernation
- dogfooding F33 changes early on F32!

Some installer pain points

- Anaconda’s default btrfs+encryption layout
  - one LUKS volume for Btrfs, one for swap
  - keeping these two + user account passwords in sync
- can’t request only first non-removable drive be wiped
  - my clowny workaround, borrowed from kickstart-list
  - this breaks ksflatten so I had to inject this snippet in
- prompting for LUKS passphrase
  - works in text mode, not so much in graphical
- wifi setup only works in graphical mode
- bug reports TBD

Other F33 Changes

- Swap on ZRAM
  - we go without a swap partition
  - ergo, only one LUKS volume
  - can’t hibernate, but hibernation does not work with Secure Boot anyway
  - when that is fixed, eh, hibernation with btrfs swapfile should work
- Nano as default
  - we’re not touching that until F33 ships :)
- Suggestion on what else to test early?

Resource Control

- Facebook’s oomd => systemd-oomd
- Btrfs does not suffer from ext4’s priority inversion
- We’re likely going to help dogfood this on the client fleet

Execution authorization

- evaluating fapolicyd
  - already deploying its macOS counterpart, Google’s Santa
- most of our custom tooling are deployed as RPMs already
- TBD:
  - gating by RPM signing keys (upstream)
  - Chef cookbook to configure fapolicyd policy and logging

Device Management

- evaluating Fleet Commander
- we prefer not to be tied to AD / FreeIPA
- MicroMDM might be a good fit
    - open source
    - one of the core contributors work for CPE
    - we use it to manage our Mac client fleet

Conclusion

Conclusion

- Desktop Linux work is increasingly XFN
  - with the Fedora community and upstream projects
  - within Facebook (prod OS, security, kernel)
- Looking to collaborate
  - with community members
  - with other companies with similar needs

Resources

- What’s new with CentOS at Facebook [CentOS Dojo]
- Upgrading CentOS on the Facebook fleet [devconf.cz]
- Facebook 💝 Fedora (and Chef) [Flock 2019]
- Facebook’s chef-cookbooks and CPE cookbooks
- The kickstarts

Q&A

facebook