Where we are - and where we go with oVirt Node

Fabian Deutsch, Red Hat
oVirt Workshop
Oct 16 2014
What it is

- “Hypervisor” for oVirt
- Minimal OS image prepared for oVirt
- Installation/upgrade on bare-metal
- Custom Installer TUI & Setup TUI

- Easy to deploy, manage and upgrade
What it is

Demo

https://www.flickr.com/photos/mrmuskrat/3637703614
Pains

● User Experience
  ○ SELinux
  ○ Installation (multipath, iscsi, efi, secureboot, …)
  ○ Customization (kmods, offline + online)
  ○ Consumption (persistence)

● Maintenance

● Build
Causes

● User experience
  ○ read-only rootfs
  ○ installation
  ○ customization

● Developers
  ○ consumption / persistence
  ○ testing

https://www.flickr.com/photos/31031835@N08/10007190363
Changes

- No custom installer
- Regular “writable” root filesystem
  Technically behaves like a regular OS
- Split rootfs, from TUI, from deployment
Installation

- Installer: Anaconda
  - Using a regular kickstart
Storage

● Specific LVM usage pattern
  ○ Boot into writable thin-volume a top a read-only “original”
Upgrade

- Add a new image + new boot entry
  - Number of upgrades is limited by disk size
  - Rollback: Select old entry, boot into old image
Local Administration

- Runtime Setup TUI is kept
  - Simplified below the surface
Improvements

★ Easier
  ○ to consume by payloads (vdsms, ...)
  ○ building + test
  ○ to extend (just a package)

★ Open to configuration management

★ Add packages/kmods
  (at runtime)
Links

- **Prosa**
  - http://dummdida.tumblr.com/tagged/node

- **Sources**
  - https://github.com/fabiand/imgbased/

- **CI**
  - https://travis-ci.org/fabiand/imgbased/
  - http://jenkins.ovirt.org/view/All/job/fabiand_ovirt-node-tng_image_build_daily/
  - http://jenkins.ovirt.org/view/All/job/fabiand_ovirt-node-tng_image_check_functional/
  - http://jenkins.ovirt.org/view/All/job/fabiand_ovirt-node-tng_image_check_install/
Open questions

- Stateless
  - Previous Node side blocker: Where to keep the state
  - Option: Generic Registration and/or Foreman
  - Diskless?

- Maximum minimization (smallest size)
  - Foreman Discovery Image