



oVirt Engine: Host Life-cycle Bootstrapping and Registering

02/11/11

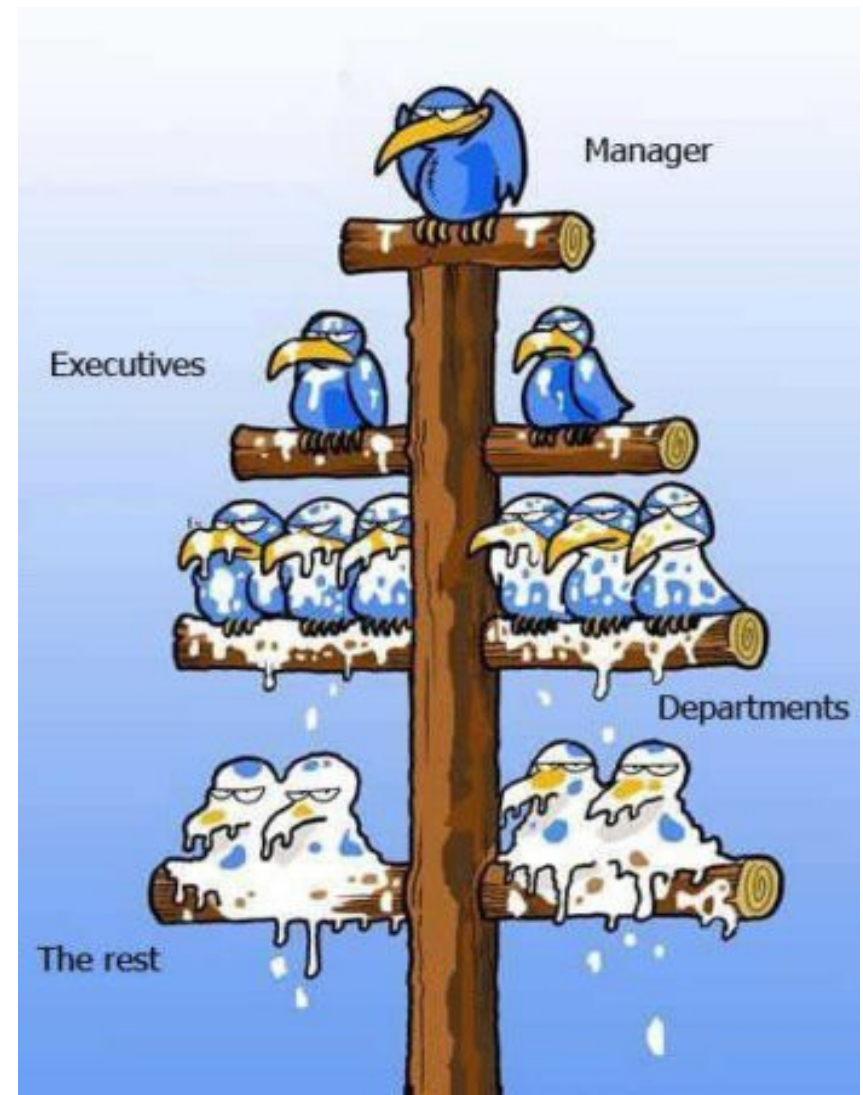
Doron Fediuck

Agenda

- Flow overview
 - Mid-level overview of the flow.
- Principle scripts
 - Map script to phase on each flow.
- Functionality- common
- Functionality- differences
- Improvements- be better!
- Code dive?

Overview

- Bootstrapping
 - The process of adding a RHEL/Fedora host to be managed by the Engine.
- Registration
 - The process of adding an oVirt node to be managed by the Engine.



Bootstrap: a Hypervisor is born!

- Initiated by the Engine side (add host)
- vds_installer.py is downloaded into the host and executed;
 - Downloads deployUtil + vds_bootstrap and run vds_bootstrap
 - Download & run vds_bootstrap_complete.
 - REBOOT

Ovirt Node registration: oVirt Hypervisor is born!



- Host needs to be configured either during manual install or PXE boot (kernel args).
 - Engine server's IP and port.
- Initiated from host side!
- In oVirt node everything is pre-installed, and only configuration is allowed.

Ovirt Node registration -continued

- vdsmd-reg daemon is doing the basic networking changes
- vdsmd-reg fetches engine public key and 'calls' the server with ID, IP and hostname.
- [Wait for admin to approve in UI or API]
- vdsmd-gen-cert generates key-pair & certificate request
- [Certificate signed in engine and downloaded back]
- vdsmd-complete runs for final touches

Adding oVirt Node as a host flow



- Huh????
 - Flow added due to users request.
- Adding an oVirt Node in the same way users add Fedora / RHEL hosts. Thus, simplifies admin's work.
- Requires the user to set root password in the node and allow SSH access to root.

Adding oVirt as a Fedora host flow -continued



- Flow starts from engine by downloading and running `vds_bootstrap.py`.
- When bootstrap identifies an oVirt node, it'll configure its registration conf file.
- Once configured, bootstrap will make sure to download the engine's public key into the host, and restart `vdsm-reg` daemon.
- This will fire the registration process with a special token.
- When engine backend now gets the registration information with the token, it'll will auto-approve the registering host.

Principle scripts



Phase	RHEL/Fedora: bootstrap	Ovirt Node: registration
Pre-action		Registration setup: <code>/usr/lib/python2.6/site-packages/ovirt_config_setup/rhevm.py</code> (manual tui script) / config-rhev-manager (PXE)
Action	vds_bootstrap	Registration service: vdsm-reg-setup
	reboot	vdsm-gen-cert
	vds_bootstrap_complete	vdsm-complete
Post-action		vdsm-upgrade

Functionality: the search for common ground



Functionality mapping



Function	RHEL: bootstrap	oVirt Node: registration
Generate vdsm.conf	vds_bootstrap	<code>/usr/lib/python2.6/site-packages/ovirt_config_setup/rhev.py</code> (manual tui script) / config-rhev-manager (PXE)
Generate engine bridge	vds_bootstrap	vdsm-reg-setup
Get engine SSH key	vds_bootstrap	vdsm-reg-setup
Time sync	vds_bootstrap	vdsm-reg-setup

Functionality mapping -continued



Function	RHEL: bootstrap	oVirt Node: registration
Generate key-pair + cert' req'	vds_bootstrap	vdsm-gen-cert
Install cert'	vds_bootstrap_complete	vdsm_complete
Setup cordump	vds_bootstrap_complete	vdsm_complete
Cleanup	vds_bootstrap_complete	vdsm_complete
Update VDSM conf from engine	vds_bootstrap_complete	vdsm_complete

Functionality: so... what's not common?



Functionality: Bootstrap only



- Platform test: handle bootstrapping oVirt Node...
- Repo test: make sure we can locate the relevant packages.
- Version test: make sure we have the right RPM.
- Virt test: make sure VT flag is on.
- OS / Kernel test.
- Packages: uninstall conflicts.
- Packages: install / update all relevant RPM's.
- Services: chkconfig conflicts off.
- Services: chkconfig relevant services on.

Functionality: Registration only

- General: persist every new file / configuration!
- vdsm-reg configuration
- Registration: post machine UUID, MAC address and other info

Improvements- be better!

- ★ Replacing http-get we do today with scp
- ★ Non root registration / bootstrapping
- ★ Thinking of a unified model to simplify implementations
- ★ *We welcome suggestions to these sensitive processes*

Code dive



oVirt

THANK YOU !

<http://www.ovirt.org>