Paul Scherrer Institute
Valeri Markushin

OS Installation and Management with Kickstart and Puppet

Community hpc-ch meeting, EPFL, Lausanne, May 20th, 2010

Overview

The goals of this talk

- to discuss the OS deployment and management tools used at PSI
- to outline main arguments and facts (contact me for details)
- to show some PSI customization features

Topics

- Scientific Linux
- Kickstart Installation
- Puppet Management
- Enhancing your Linux distribution

Requirements for Operating System

Our requirements for HPC at PSI are (mostly) met by Scientific Linux (Red Hat), http://www.scientificlinux.org/.

Scientific Linux (RHEL)

- open source
- meet the requirements of big user communities (HEP, . . .)
- long–term support
- good HW support (servers, workstations)
- good commercial SW support

Our Linux path over the past 10 years:
. . . → RH7 → RH9 → ? → SL3 → SL4 → SL5 **(SL54)** → SL55
Disclaimer: SL(RH) may not be your best choice for notebooks and multimedia desktops.

Requirements for Installation and Management

Systems installed and managed

- **HPC and dedicated clusters** (Linux)
- servers (Linux, Solaris)
- high–end workstations (Linux)
- desktops (Linux or dual boot)
- notebooks (Linux or dual boot)
- total of 600 and growing

Unified installation and update mechanism

We want to use the same system management framework for all hosts through their complete life cycle.

Linux Kickstart Installation

Kickstart installation method from Red Hat: **Anaconda** installer.

The Kickstart installation tasks

- define the host identity (HOSTNAME, . . .)
- partition the disk(s)
- configure the primary NIC
- install basic RPMS
- some other helpful things

Install once, update many times — **keep the installer simple and fast, do the most of host customization later.**

Kickstart Customization

The Kickstart customization

- very few **kickstart configuration files** (server, cnode, desktop, 64- or 32-bit)
- many **customization keys** to specify the installation and the host properties in more details (optional, PSI specific)
- fully automatic installation is possible
- exact replication of an installation is possible

Example of PXELinux boot

```
boot: sl5c hostid=lchpc99:Cnode/lchpc
   sl5c --- PXELinux label (KS configuration)
   cnode42 --- HOSTNAME
   Cnode/lchpc --- customization key
```

Kickstart Customization Keys

Anaconda installer

- pre-installation (user input if needed)
- Anaconda core (keep it small)
- post-installation, nochroot
- post-installation, chroot

Kickstart customization keys (PSI extension)

- an array of strings defined at the kickstart
- control the logic of the pre– and post–installation scripts
- provide an interface to Puppet via a predefined set of environment variables

Puppet

Puppet is a systems management framework from Puppet Labs: http://www.puppetlabs.com/.

Puppet features

- **declarative** language describing the system configuration
- enforcing the desired system configurations
- graph based model of hosts declaration
- modular design, easily extensible
- idempotent
- cross-platform management (Linux, Solaris, . . .)

The current version: **0.25.1** → 0.25.4.

Puppet Tasks

Everything is a file, and **Puppet** is our files management tool.

Puppet tasks

- get Puppet **facts** for the host
- install and update the packages according to the host configuration
- provide the required configuration files
- configure services
- configure environment modules, users, etc.

The goal is to ensure the desired system configurations.

Puppet Facter

Information about a system is provided to **puppet** by **facter**, which is easy to expand and well suited for **customization**.

```
# facter -p
arch => x86_64
auto_update => yes
auto_update_rpms => yes
auto_update_users => yes
...
ipaddress_eth0 => 129.129.199.199
hostname => cnode42
kernelrelease => 2.6.18-164.15.1.el5
role => cnode
set => hpc
zone => psiwest
...
```

Puppet at PSI

Puppet configuration features at PSI

- puppet master servers: 2 production and 1 testing
- the puppet environment: separate **environments** for different host groups (organizational and functional)
- each puppet environment uses its own definition of nodes, the modules being (partly) shared
- each puppet environment has its own principal maintainer

/etc/puppet/puppet.conf

```
[main]
    ...
    environment = sl5c
[puppetd]
    server = puppet1.psi.ch
    ...
```

Managing files with Puppet

cfile(path) extending file(path)

Use a predefined array of paths based on the **zone–set–role–hostname** hierarchy (provided by **facter**) to extend the functionality of the file resource.

```
cfile("/etc/hosts")
/var/puppet/environments/sl5c/modules/custom/files
|-- psiwest                          zone
    |-- etc/hosts                    4
    |-- hpc                          set
        |-- etc/hosts                3
        |-- cnode                    role
            |-- etc/hosts            2
            |-- cnode42              hostname
                |-- etc/hosts        1
```

Additional (non-SL/RH) Software

The version of XXX provided by SL(RH) is too old!
What should we do?

Environment Modules

Additional software is mainly installed as **environment modules**, http://modules.sourceforge.net/.

- clean process environment
- more than 70 products, multiple versions
- no conflicts with the SL(RH) packages
- customizable (system, user)

No local installation — only symlinks:

```
/opt/name/version -> /afs/.../opt/name/version
```

Services

Services needed for Kickstart and Puppet

- **DHCP** server
- **TFTP** server for PXELinux
- **HTTP** server
  - the KS installation tree
  - **kickstart configuration** files and scripts
  - the YUM repositories
- **Puppet master** server(s)

Other services

- AFS: home directories on desktops
- AFS: software installed as environment modules
- GPFS: home and scratch directories on clusters

Summary

PSI Experience Summary

- mostly conventional tools (kickstart, Puppet) without modification
- kickstart customization keys — simple things are easy
- kickstart–Puppet interface — install once, update many times
- customized Puppet manifests — get exactly what your systems need
- hierarchical hosts configuration works well
- expected to be scalable to thousands hosts

Acknowledgments

Many thanks to my colleagues at AIT, PSI for the fruitful collaboration on the Linux installation and management systems: Heiner Billich, Derek Feichtinger, Marc Gasser, Rene Kapeller.