

Connexer Ltd.

Technology Consulting

Using `git` To Manage Your System's Configuration

Roberto C. Sánchez
Ohio LinuxFest
Saturday, October 25, 2014

Overview

- About the presenter
- Use case
- Alternatives for managing system configuration
- `git`-based alternatives
- Selecting an approach
- Implementing your solution
- Demonstration

About the Presenter

- I am a software engineer and IT consultant
- I run my own one-man consulting firm
- I have been using Linux since 2002
- I have been a Debian Developer since 2007
- I am online here:
 - <http://people.connexer.com/~roberto>
 - <http://www.connexer.com>
 - <http://www.linkedin.com/in/robertocsanchez/>

Use Case

- *Very* small sysadmin team ($1 \leq n \leq 3$)
- Not full-time configuration managers
- Possibly not even full-time sysadmins
- Small number of systems ($1 \leq n \leq 20$)
- Heterogeneous systems
- Virtual machines or real machines

Alternatives for Managing System Configuration

- Do nothing
 - Example: *[nothing]*
 - Pro: no additional effort up front
 - Con: lots of headaches if anything goes wrong
- Use **cp**
 - Example: `cp /etc/fstab /etc/fstab.bak.9`
 - Pro: minimal up front effort
 - Con: error prone (c.f., why devs use VCS)

Alternatives for Managing System Configuration

- Let your distro's package system manage it
 - Example: *.dpkg-dist/.dpkg-new/.dpkg-old*
 - Pro: Well tested and already works
 - Con: Doesn't help for admin-driven changes
- Use a purpose-built CM solution
 - Example: CFEngine, Puppet, Chef, etc.
 - Pro: Extremely powerful and flexible
 - Con: Massive overkill for “small” situations

Alternatives for Managing System Configuration

- Use a manual VCS-based solution
 - Example: Check entire `/etc` tree into CVS or SVN
 - Pro: Going “back in time” becomes possible
 - Con: Unintegrated, possibly sub-optimal workflow

git-based Alternatives

- Use straight Git
 - Example: Check entire `/etc` tree into Git
 - Pro: Leverage the power of Git (or any DVCS)
 - Con: Not integrated with system events
- Use a wrapper like `etckeeper`
 - Example: Well, `etckeeper`
 - Pro: Power of DVCS integrated w/ system events
 - Con: Adds an extra “layer”

etckeeper package description

- “etckeeper is a collection of tools to let `/etc` be stored in a git, mercurial, darcs, or bazaar repository. It hooks into apt (and other package managers including yum and pacman-g2) to automatically commit changes made to `/etc` during package upgrades. It tracks file metadata that revision control systems do not normally support, but that is important for `/etc`, such as the permissions of `/etc/shadow`. It's quite modular and configurable, while also being simple to use if you understand the basics of working with revision control.”
- <http://joeyh.name/code/etckeeper/>

Selecting An Approach

- Choose a solution that fits for the whole team
- If the approach does not dictate a workflow, develop a suitable workflow for the team
- For the purposes of this presentation, let's assume that we have chosen etckeeper

Selecting An Approach

- Reasons to use a VCS-based solution:
 - Many admins code and are familiar with VCS
 - “Automates” the documentation process
 - Acts as an organic backup of system configuration
 - Facilitates troubleshooting and fault isolation
 - Makes multi-admin collaboration less error-prone
 - Many other advantages ...
- Be mindful of security issues
 - Potential leakage of sensitive information

Implementing Your Approach

- Install required packages:
`sudo apt-get install etckeeper`
- Configure `git`:
 - `git config --global user.email "abc@xyz.com"`
 - `git config --global user.name "John Doe"`
- Read the documentation:
`zless /usr/share/doc/etckeeper/README.gz`
- Be as hands-on or hands-off as you like

Summary

- Using Git (or any VCS) saves headaches
 - Makes troubleshooting easier
 - Makes collaboration easier
 - Makes managing your system(s) easier
- There are some potential pitfalls
 - Potential leakage of sensitive information
 - Potential additional complexity

Questions?