
# Atomic Updates and /etc

## How to handle update of configuration files

Thorsten Kukuk
Distinguished Engineer
Sr. Architect SLES & MicroOS
Future Technology Team
kukuk@suse.com

# Background

# RPM and Configuration Files

- **RPM has support for configuration files**
  - files not marked: an update would replace the file and all changes are lost
  - files marked as %config: modified files are moved away as *.rpmsave
  - files marked as %config(noreplace): modified files stay, new files are written as *.rpmnew
- **RPM support for configuration files hasn’t changed since many, many years**
- **So everything must be Ok**

# Really?

**After every update, the sysadmin has to:**

- Search for *.rpmsave and *.rpmnew files
  - Thanks for all the fixed typos in comments, which moved my changed configuration files to *.rpmsave and broke the services!
- Merge all changes manually

**If not: the services can be broken, insecure, ...**

# Atomic Updates

## An atomic update is a kind of update that:

- Is atomic
  - Either fully applied or not at all
  - The update does not influence your running system
- Can be rolled back
  - If the update fails or if the update is not compatible, you can quickly restore the situation as it was before the update

# Atomic Updates and Configuration Files

## With Atomic Updates, it’s even getting worse:

- Admin applies update
  - This is not visible until next reboot, including changes to configuration files
- Admin adjusts current (old) configuration files
  - Maybe the atomic update also made changes not yet visible?
- Admin reboots
  - What should happen with the modified configuration files?
  - How can the admin find out:
    - Which files are modified?
    - Which changes were made?

# Factory Reset of systemd

## http://0pointer.net/blog/projects/stateless.html

- A mechanism we call Factory Reset shall flush out /etc and /var, but keep the vendor-supplied /usr, bringing the system back into a well-defined, pristine vendor state with no local state or configuration. This functionality is useful across the board from servers, to desktops, to embedded devices.

## No major Linux Distribution did really implement this

- Requires strict following of FHS for /etc and /var
  - snapshot+rollback and transactional-update require the same
  - openSUSE nearly has this for /var
  - openSUSE is far away with /etc for this

# We are not alone with the problem

- Most Linux Distributors
  - have a distribution with atomic updates
  - facing the same problem
  - have a different solution

- “Upstream”
  - Current code works for them
  - Are not interested in Linux Distribution specific patches/solutions

- Users
  - Don’t know where to find the configuration file on this distribution → Pre-FHS!

# Goal

## Provide a concept working for all Linux Distributors like FHS

- Users should find the configuration always in the same place
- Guidance for new software development and packager
- Short to mid term: solve atomic update problems
  - Concentrate on packages for a Container Host OS
  - Enhance to full distribution
- Long term: /etc only contains changes made by the admin

## It’s not:

- Requirement for all packages now
- Shuffle everything around which has alreaday a solution

# Requirements for a solution

**Define a new way to store and manage configuration files, where:**

- It's visible to the admin that something got updated
- It's visible, which changes the admin made
- Best if the changes could be merged automatically

# Proposals

# Application configuration files

**Do it similar to systemd:**

- look for /etc/application.conf
- if it does not exist, load /usr/share/defaults/application/application.conf
- look for overrides in /etc/application.conf.d and merge them

**See https://www.freedesktop.org/software/systemd/man/systemd.unit.html#Examples, "Overriding vendor settings" for more details and examples.**

# Sounds like a lot of coding work?

**Many applications have already support for this, but we don’t use it:**

- PAM → /usr/lib/pam.d, /etc/pam.d
- sysctl → /usr/lib/sysctl.d/*.conf, /etc/sysctl.d/*.conf → drop /etc/sysctl.conf
- ldconfig → Move ld.so.conf and use e.g.:
  - /usr/share/defaults/ldconfig/ld.so.conf.d for distribution specific files, e.g. libgraphivz6
  - /etc/ld.so.conf.d for local changes

**Several more tools have .d directories, enhance them:**

- aliases.d, ant.d, bash-completion.d, chrony.d, cron.d, depmod.d, dnsmasq.d, dracut.conf.d, grub.d, issue.d, logrotate.d, modprobe.d, netconfig.d, sudoers.d, …

# System databases

**Some files in /etc are strictly spoken no configuration files, like:**

- /etc/rpc
- /etc/services
- /etc/protocols

**Move them to *<u>/usr/share/defaults/etc</u>***

**/etc/{rpc,services,protocols} contains only additional local entries**

**libnss_usrfiles will look first in /etc, afterwards in /usr/share/defaults/etc**

**Can be done already today**

# /etc/passwd, /etc/group, /etc/shadow

**No good idea/solution up to now**

**Idea:**

- System accounts in /usr/share/defaults/etc/{passwd,group,shadow}
- Normal accounts and changes in /etc/{passwd,group,shadow}
- glibc NSS plugin to read files from both locations and merge them

**Drawbacks:**

- Does not solve all  problems (admin wants to create system accounts)
- nss_compat does not work anymore
- Confusing: passwords for account in /etc/shadow, but not in /etc/passwd

# Default location

# Where to store default configuration files

**As there is not yet a standard directory below /usr, a new one needs to be created. There are some requirements:**

- Easy to find and remember for the system administrator
  - Best: all in one place
  - Don’t clobber /usr/lib even more, that’s not easy to find
- No conflict with FHS
- The name should not confuse administrators
- It should be clear, that this are default configuration files and changes should not be done here

# Already in use or suggested directories

- /usr/share/defaults/{etc,skel,ssh,ssl}: ClearLinux and several packages
- /usr/share/{baselayout,skel,pam.d,coreos,...}: CoreOS/Container Linux
- /writeable,/etc/writeable: Ubuntu Core
- /usr/etc: openSUSE MicroOS, RedHat/Fedora/CentOS Atomic
- /usr/share/sysconfig: but different meaning than /etc/sysconfig
- /usr/share/misc: used by several tools already, but not really compatible with FHS

# My current favorite

- /usr/share/defaults - contains everything, which else would be belong to /etc
- /usr/share/defaults/etc - aliases, ethers, protocols, rpc, services
- /usr/share/defaults/etc - shells, ethertypes, network: copied with systemd-tmpfiles
- /usr/share/defaults/skel - systemd-tmpfiles will symlink this files into /etc/skel
- /usr/share/defaults/pam.d - default distribution specific PAM configuration files. /etc/pam.d will overwrite this.
- /usr/share/defaults/<application> - application specific files, read directly or copied to /etc via systemd-tmpfiles
- /usr/*/<application> - application specific files, can include configuration files
- /usr/lib/sysimage/etc - passwd, group, shadow containing system users

# Questions?

**Thank you.**

**The full proposal:**
**https://github.com/thkukuk/atomic-updates_and_etc/blob/master/README.md**