oVirt 3.3 Packaging

Alon Bar-Lev
Red Hat

Alonbl's Time line

- 3.1:
  - Re-wrote SSH
  - Reduced the differences between upstream and downstream packaging.
- 3.2
  - Re-wrote host-deploy (except of vdsm-reg).
  - Cleanup PKI (PKCS#12, separate web trust chain).
  - Introduce the minor version into all projects but vdsm.
- 3.3
  - Re-wrote packaging.
  - Re-wrote PKI.
  - Introduction of production like development environment.
  - Complete SSH features (ybronhei)
  - Backup/restore (oschreib)
  - Rename host.

Packaging

- Build system
- RPM packaging
- Setup, upgrade and clean
- Services' wrappers
- Helper programs

What's new?

- Source tree re-org
- Build system rewrite (except java)
- RPM packaging interface
- PKI scripts rewrite
- Log Handling
- Database credentials
- Service wrappers rewrite
- Setup, upgrade, cleanup rewrite
- Apache integration

What's new?

- Backup/restore utility
- Host rename utility
- Branding, osinfo
- Development environment
- Gentoo support

Source tree re-org

- Everything is under ./packaging
- File names much installation names.
- Structure is more or less the same as target system layout.
    - bin
    - branding
    - conf
    - dbscripts
    - etc
    - firewalld
    - man
    - pki
    - pythonlib
    - services
    - setup
    - sys-etc

Build system rewrite

- Why?
  - No interface.
  - No customization.
  - No best practices.
- How?
  - Generic Makefile to copy directory structure and not specific files.
  - Formalized interface as variables, example:
    - BUILD_GWT=1
    - BUILD_GWT_WEBADMIN=0
  - Support development environment installation.

RPM Packaging Interface

- Why?
  - RPM was built using different commands and parameters.
  - Internal change effected all builders.
- How?
  - Formalize interface to ease CI
    - ovirt_build_quick
    - ovirt_build_minimal
    - ovirt_build_gwt
    - ovirt_build_locales

PKI scripts rewrite

- Why?
  - Too many bugs.
  - Unusable for support.
- How?
  - File location and format remained as-is to avoid breaking backward compatibility.
  - Human usable interface, example:

```
$ pki-enroll-pkcs12.sh --help
Usage: pki-enroll-pkcs12.sh [OPTIONS]
Generate key, enroll certificate, store in PKCS#12 format.
Result will be at /etc/pki/ovirt-engine/keys/PREFIX.p12

    --name=prefix               file name without prefix.
    --password=password password of PKCS#12.
    --subject=subject           X.500 subject name.
    --keep-key                  reissue certificate based on previous request.
```

Log Handling

- Why?
  - Remove proprietary functionality.
  - Apply a policy to all logs.
- How?
  - Logs are handled using logrotate instead of proprietary rotation.
  - All logs are managed.

Database Credentials

- Why?
  - Obscuration is actually less secure.
  - Too many places and formats credentials were stored.
  - We cannot assume we have dba premision.
- How?
  - No use/store database privileged account (dba)
    - During setup if user choose to automatically configure database we create user and empty database on behalf of the user.
  - pgpassfile format is not used any more.
  - Database password obscuration is not used any more.
  - Database credentials are stored within file readable only by ovirt user. File is: /etc/ovirt-engine/engine.conf.d/10-setup-database.conf

Services' Wrappers Rewrite

- Why?
  - Tightly coupled with sysv.
  - No use of downstream services.
  - Some behavior is not standard.
- How?
  - Use downstream services (sysv, systemd) properly
  - Reduce the logic of our implementation in favor of downstream specific functionality.
  - Use common infrastructure for all services.
  - Expose common infrastructure for reuse by other distributions.

Setup, upgrade, cleanup rewrite

- Why?
  - Low code quality.
  - RHEL/Fedora specific.
- How?
  - Implementation using otopi
    - Pluggable installation sequence framework
    - Shared with host-deploy
    - Modular implementation
    - Environment based
  - Transactional (when legacy permits)
  - Excessive logging
  - Enhanced user dialog
  - Automation friendly
  - Uninstall friendly
  - Platform independent

Modular Layout

setup/bin
setup/dbutils
setup/ovirt_engine_setup
setup/plugins
setup/plugins/ovirt-engine-common
setup/plugins/ovirt-engine-common/core
setup/plugins/ovirt-engine-common/db
setup/plugins/ovirt-engine-common/dialog
setup/plugins/ovirt-engine-common/distro-rpm
setup/plugins/ovirt-engine-common/system
setup/plugins/ovirt-engine-remove
setup/plugins/ovirt-engine-remove/config
setup/plugins/ovirt-engine-remove/core
setup/plugins/ovirt-engine-remove/db
setup/plugins/ovirt-engine-remove/files
setup/plugins/ovirt-engine-setup
setup/plugins/ovirt-engine-setup/all-in-one
setup/plugins/ovirt-engine-setup/apache
setup/plugins/ovirt-engine-setup/config
setup/plugins/ovirt-engine-setup/core
setup/plugins/ovirt-engine-setup/db
setup/plugins/ovirt-engine-setup/dialog
setup/plugins/ovirt-engine-setup/distro-rpm
setup/plugins/ovirt-engine-setup/legacy
setup/plugins/ovirt-engine-setup/network
setup/plugins/ovirt-engine-setup/pki
setup/plugins/ovirt-engine-setup/provisioning
setup/plugins/ovirt-engine-setup/releasepreview
setup/plugins/ovirt-engine-setup/system
setup/plugins/ovirt-engine-setup/upgrade

Development environment

- Why?
  - Developers actually abused product to make it work.
  - The gap between their environment and production was huge.
  - Any change in product resulted in breaking the environment as it was installed against product usage.
- How?
  - Direct outcome of setup rewrite.
  - Installation of product using unprivileged user.
  - All services are enabled:
    - tools (config, domains)
    - PKI
    - host-deploy
  - README.developer, wiki

Apache Integration

- Why?
  - Supporting multiple configuration is difficult.
  - Currently we assume we own apache.
- How?
  - Force apache usage.
  - Proxy only ovirt-engine URIs, to allow sharing apache with other applications.
  - Optionally redirect root URI to product, to support site specific welcome page.
  - ovirt-engine URI is /ovirt-engine/

Branding and osinfo

- Why?
  - Allow product customization post install.
  - Reduce difference between upstream and downstream.
- How?
  - Use drop dir methodology.
    - /etc/ovirt-engine/branding/
    - /etc/ovirt-engine/osinfo.conf.conf/
  - Packager friendly, easy to extend.

Gentoo support

- Why?
  - Extend community.
  - The most complex and flexible packaging as preparation for other distributions.
- How?
  - ovirt-engine only!
  - Gentoo Overlay
    - https://github.com/alonbl/ovirt-overlay
  - Gentoo Instructions
    - http://wiki.gentoo.org/wiki/OVirt

Summary

- Packager friendly.
- Distribution friendly.
- DBA friendly.
- Ability to share apache, postgresql with other applications.
- Pluggable and extensible.