Software Collections

Honza Horak <hhorak@redhat.com>

hhorak @ freenode
Red Hat, Brno
CentOS Dojo 30th Jan 2015

All versions of any software on your system. Together.

Software management challenges

- Enterprise systems:
  - Single version of software
  - Not break stuff => no changes in base system
- Users want newer versions
  - New features, testing
  - Apps require specific version that is not in OS
- System providers need to offer more versions

We believe it is possible

- ..or at least we didn't know it was not :)
- Focus on RPM world
  - Easy for packagers
  - Easy to use
  - RPM pros and cons

Software Collections principles

Concepts of Software Collections (SCL)

- Stacks rather then packages/libraries
  - Python 3.3, MariaDB 10.0, ...
  - *Meta package* enables the stack
  - Missing deps are included into collection
- Collections naming conventions
  - Use “major” version
  - Examples: python33, mariadb100

Concepts of Software Collections (SCL)

- Using /opt, /etc/opt, /var/opt
  - And non-conflicting names
    - RPM metadata
    - Files outside /opt (e.g. SysV init script)
- Using *vendor* under /opt
  - For comply with File Hierarchy Standard
  - Example: /opt/rh, /opt/fedora, /opt/centos

Technology behind SCL

- *scl-utils* available in CentOS/RHEL
  - Building support (*scl-utils-build*)
  - Run-time support
  - https://github.com/sclorg/scl-utils

Meta package overview

postgresql92

- Metapakage (empty), handles dependencies

postgresql92-runtime

- /etc/scl/prefixes/postgresql92
- /opt/<vendor>/postgresql92/enable
- /opt/<vendor>/postgresql92/root/usr/bin...

postgresql92-build

- /etc/rpm/macros.postgresql92-config
  - (macros for collection)

postgresql92-scldevel

- /etc/rpm/macros.postgresql
  - (macros for depended collections)

Packages of PostgreSQL 9.2 SCL

```
postgresql92-0:1.1-20.el6.x86_64
postgresql92-build-0:1.1-20.el6.x86_64
postgresql92-runtime-0:1.1-20.el6.x86_64
postgresql92-scldevel-0:1.1-20.el6.x86_64

postgresql92-postgresql-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-contrib-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-devel-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-docs-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-libs-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-plperl-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-plpython-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-pltcl-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-server-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-test-0:9.2.8-1.el6.x86_64
postgresql92-postgresql-upgrade-0:9.2.8-1.el6.x86_64
```

Example of content

/etc/pam.d/postgresql92-postgresql

/etc/rc.d/init.d/postgresql92-postgresql

/etc/opt/rh/scls/postgresql92/sysconfig/postgresql

/opt/rh/postgresql92/root/usr/bin/initdb

/opt/rh/postgresql92/root/usr/bin/pg_ctl

/opt/rh/postgresql92/root/usr/bin/postgres

/opt/rh/postgresql92/root/usr/bin/postmaster

/opt/rh/postgresql92/root/usr/lib64/pgsql/ascii_and_mic.so

/opt/rh/postgresql92/root/usr/lib64/pgsql/euc2004_sjis2004.so

/opt/rh/postgresql92/root/usr/share/man/man1/initdb.1

/opt/rh/postgresql92/root/usr/share/man/man1/pg_controldata.1

/opt/rh/postgresql92/root/usr/share/pgsql/contrib

/opt/rh/postgresql92/root/usr/share/pgsql/conversion_create.sql

/var/opt/rh/scls/postgresql92/lib/pgsql/backups

/var/opt/rh/scls/postgresql92/lib/pgsql/data

/var/lib/pgsql

Example of install

```
#> yum install postgresql92
...get a coffee :) ...
#> rpm -qa postgresql92*
postgresql92-1.1-20.el6.x86_64
postgresql92-postgresql-9.2.8-1.el6.x86_64
postgresql92-postgresql-libs-9.2.8-1.el6.x86_64
postgresql92-postgresql-server-9.2.8-1.el6.x86_64
postgresql92-runtime-1.1-20.el6.x86_64
```

Content of *enable* scriptlet

```
export PATH=/opt/rh/postgresql92/root/usr/bin$
{PATH:+:${PATH}}

export LIBRARY_PATH=/opt/rh/postgresql92/
root/usr/lib64${LIBRARY_PATH:+:${LIBRARY_PATH}}

export LD_LIBRARY_PATH=/opt/rh/postgresql92/
root/usr/lib64${LD_LIBRARY_PATH:+:
${LD_LIBRARY_PATH}}

export MANPATH=/opt/rh/postgresql92/root/
usr/share/man:${MANPATH}
```

Example of running

```
#> scl enable postgresql92 'psql --version'
psql (PostgreSQL) 9.2.8

#> psql --version
psql (PostgreSQL) 8.4.20

#> service postgresql start
#> service postgresql92-postgresql start

#> systemctl start postgresql92-postgresql
```

Live demo:
python-dateutil into python33 SCL

Live demo with Python33

- Build python-dateutil On CentOS 7
- Meta package overview
- How to build collections in mock (in epel)
- http://goo.gl/JXT6k2

  -
    ```
    yum install mock rpmdevtools spec2scl
    ```

  -
    ```
    usermod -a -G mock
    ```

  -
    ```
    curl https://hhorak.fedorapeople.org/python33-scl-el7/centos7-python33.cfg -o /etc/mock/centos7-python33.cfg
    ```

  -
    ```
    mock -r centos7-python33 --init
    ```

Example of spec2scl output (after correction)

```
@@ -1,6 +1,9 @@
+%{?scl:%scl_package python-dateutil}
+%{!?scl:%global pkg_name %{name}}
-Name:           python3-dateutil
+Name:           %{?scl_prefix}python-dateutil
-BuildRequires:  python3-devel,python3-setuptools,python3-six
-Requires:       tzdata,python3-six
+BuildRequires:  %{?scl_prefix}python-devel
+BuildRequires:  %{?scl_prefix}python-setuptools
+BuildRequires:  %{?scl_prefix}python-six
+Requires:       tzdata
+Requires:       %{?scl_prefix}python-six
 %build
+%{?scl:scl enable %{scl} - << \EOF}
 %{__python3} setup.py build
+%{?scl:EOF}
 %install
+%{?scl:scl enable %{scl} - << \EOF}
 %{__python3} setup.py install --skip-build --root $RPM_BUILD_ROOT
+%{?scl:EOF}
-%files -n python3-dateutil
+%files -n %{?scl_prefix}python-dateutil
 %doc example.py NEWS README
 %license LICENSE
 %{python3_sitelib}/dateutil/
```

Example of mock config

```
# cat /etc/mock/centos7-python33.cfg
config_opts['root'] = 'centos7-python33'
config_opts['target_arch'] = 'x86_64'
config_opts['legal_host_arches'] = ('x86_64',)
config_opts['chroot_setup_cmd'] = 'install rpm-build "@Development Tools" scl-utils-build python33-build'
config_opts['dist'] = 'centos7'  # only useful for --resultdir variable subst

...

[python33repo]
name=python33repo
baseurl=file:///home/hhorak/python33-scl-el7/
enabled=1
```

Advanced Software Collections packaging

SCL-izing dynamic languages

- Language stacks use advanced RPM features
  - require/provide generators
  - well-known macros %{python_sitelib}
  - %{scl_package_override}

e.g. perl is not simple at all

```
%scl perl520
%nfsmountable 1

%tests_req() %{?perl520tests_req}%{!?perl520tests_req:%{expand:\
BuildRequires: %* \
%%tests_subpackage_requires %* \
}}

%tests_subpackage_requires() %{?perl520tests_subpackage_requires}%{!?perl520tests_subpackage_requires:%{expand: \
%global perl520__tests_spkg_req %{?perl520__tests_spkg_req} %* \
}}

%perl_default_subpackage_tests %{?perl520perl_default_subpackage_tests}

%scl_package_override() %{expand: \
%global perl_small 1 \
%global perl_bootstrap 1 \
%global __perl LD_LIBRARY_PATH="/opt/rh/rh-perl520/root/usr/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}"
%_scl_root/usr/bin/perl \
%global __perl_requires /usr/lib/rpm/perl.req.stack \
%global __perl_provides /usr/lib/rpm/perl.prov.stack \
...
```

SCL-izing daemons

- Some files placed outside of /opt
  - SysV service file
  - systemd unit file
  - `/etc/rc.d/init.d/<scl>-daemon`
- Daemon run in clean environment
  - `source scl_source enable <scl>`
  - `ExecStart=/usr/bin/scl enable <scl> -- /opt/<ven>/root/usr/bin/daemon arg`
- SELinux
  - `semanage fcontext -a -e / /opt/<ven>/root/<scl>`

How to extend SCL

- Two ways:
  - Build new package to the existing collection
  - Create depended collection

How depended collections work

- It is quite normal collection
- Difference in enable scriptlet:
  - e.g. collection <B> depends on collection <A>
  - As soon as <B> is enabled, <A> is enabled as well
- Simple in enable scriptlet:
  - `source scl_source enable <B>`

http://developerblog.redhat.com/2014/12/04/add-packages-to-python-2-7-software-collection/

So the concept exists, but...

- How to get collections in CentOS?
- Where to join upstream to:
  - Help fixing bugs in collections?
  - See the near future?
  - Influence development?
  - Talk to developers?
- **How to build an application on top of SCL?**

Where is CentOS?

CentOS is important for SCL

- SCL is popular, but community not big yet
- Fedora proofed to be slower to adopt SCLs
- OpenShift and other would like to use SCLs in CentOS
- SCLo SIG established
- Current status
  - Infrastructure almost ready
  - Now tweaking for SCLs

SCL work-flow vision

1. become a **member of SCLo** SIG
   - get access to git under sclo project (namespace)
2. develop/**contribute** to a collection
   - under sclo/ namespace in git.centos.org
   - any git structure here (fedora-like), playground, testing
   - build collection from SRPMs (cbs.centos.org, copr.fedoraproject.org)
3. make **SCL official**
   - move SRPM from sclo/ or RHSCL into rpms/ git namespace
4. **build** the collection from git in cbs.centos.org

Questions?

https://www.softwarecollections.org/en/docs/guide

http://wiki.centos.org/SpecialInterestGroup/SCLo

Community mailing-list: sclorg@redhat.com

These slides: http://goo.gl/ZAsGjt

Honza Horak

e-mail: hhorak@redhat.com

hhorak @ freenode

Twitter: @HorakHonza