



# Software Collectoins for bleeding edge stacks on enterprise

Honza Horak <[hhorak@redhat.com](mailto:hhorak@redhat.com)>

[hhorak @ freenode](#)

Twitter: [@HorakHonza](#)

Fosdem 1<sup>st</sup> Feb 2015

All versions of any software  
on your system. Together.



# We believe it is possible

- ..or at least we didn't know it was not :)
- And btw. this is not about containers...



# Software management challenges

- Enterprise systems:
  - Single version of software
- Service providers need to offer more versions
- Users want newer versions
  - New features, testing
  - 3<sup>rd</sup> party apps require specific version not in OS
  - Not break stuff => **no changes in base system**



# Borders set up

- Focus on RPM world
  - Easy for packagers
  - Easy to use
  - RPM pros and cons



# Software Collections principles



# What packages are part of the collection

- Packages we want to use
  - Stacks rather than single packages/libraries
  - Ruby on Rails, PostgreSQL 9.2 with extens.
- Missing deps not available in base system
- *Meta package* that defines and describes the stack



# Meta package overview

## **postgresql92**

- Main metapackage (empty), pulls dependencies

## **postgresql92-runtime**

- Provides basic root filesystem under /opt

## **postgresql92-build**

- Macros for building packages for this collection

## **postgresql92-scldevel**

- RPM macros for building depended collections





# What is the whole magic?

- Avoid conflict with base system on:
  - packages name level
  - filesystem level
  - RPM metadata (provides, requires) level



# Packages of python33 SCL

postgresql92-1.1-20.el7

postgresql92-build-1.1-20.el7

postgresql92-runtime-1.1-20.el7

postgresql92-scldevel-1.1-20.el7

postgresql92-postgresql-9.2.8-2.el7

postgresql92-postgresql-contrib-9.2.8-2.el7

postgresql92-postgresql-devel-9.2.8-2.el7

postgresql92-postgresql-docs-9.2.8-2.el7

postgresql92-postgresql-libs-9.2.8-2.el7

postgresql92-postgresql-plperl-9.2.8-2.el7

postgresql92-postgresql-server-9.2.8-2.el7

postgresql92-postgresql-upgrade-9.2.8-2.el7

postgresql92-boost-1.54-1.el7



# What is the whole magic?

- Avoid conflict with base system on:
  - **packages name level**
  - filesystem level
  - RPM metadata (provides, requires) level



# Example of content

```
#> rpm -ql postgresql92-postgresql-server
/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
/var/opt/rh/scls/postgresql92/lib/pgsql/backups
/var/opt/rh/scls/postgresql92/lib/pgsql/data
/var/lib/pgsql
[...]
```



# What is the whole magic?

- Avoid conflict with base system on:
  - **packages name level**
  - **filesystem level**
  - RPM metadata (provides, requires) level



# Example of provides

```
#> rpm -q --provides postgresql92-postgresql-server
postgresql92 = 1.1-20.e17
postgresql92(x86-64) = 1.1-20.e17
postgresql92-build = 1.1-20.e17
postgresql92-build(x86-64) = 1.1-20.e17
postgresql92-postgresql = 9.2.8-2.e17
postgresql92-postgresql(x86-64) = 9.2.8-2.e17
[...]
```



# What is the whole magic?

- Avoid conflict with base system on:
  - **packages name level**
  - **filesystem level**
  - **RPM metadata (provides, requires) level**



# Technology behind SCL

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





# Example of install SCL

```
#> 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
```



# Example of using SCL

```
#> psql --version
```

```
psql (PostgreSQL) 8.4.20
```

```
#> scl enable postgresql92 'psql --version'
```

```
psql (PostgreSQL) 9.2.8
```



# How come?

```
#> scl enable postgresql92 bash
```

```
#> env
```

```
MANPATH=/opt/rh/postgresql92/root/usr/share/man:
```

```
LD_LIBRARY_PATH=/opt/rh/postgresql92/root/usr/lib  
64
```

```
PATH=/opt/rh/postgresql92/root/usr/bin:/usr/local  
/sbin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
PKG_CONFIG_PATH=/opt/rh/postgresql92/root/usr/lib  
64/pkgconfig
```

```
[...]
```



# Example of using SCL daemons

```
#> service postgresql start
```

```
#> service postgresql192-postgresql start
```

```
#> systemctl start postgresql192-postgresql
```



# New features in Software Collections

- Use collection if `/opt` is mounted:
  - `scl register /path/to/scl`
- environment modules support
  - `module load python33`



# Packaging SCL is simple



# Simple meta package spec file

```
%global scl mariadb100
%scl_package %scl

Name: %scl_name
Requires: %{scl_prefix}mariadb-server
BuildRequires: scl-utils-build
?
[...]
```

```
%install
%scl_install
?
@at >> %{buildroot}%{_scl_scripts}/enable << EOF
export PATH=%{_bindir}\${PATH:+:\${PATH}}
export LD_LIBRARY_PATH=%{_libdir}\${LD_LIBRARY_PATH:+:\${LD_LIBRARY_PATH}}
export MANPATH=%{_mandir}:\$MANPATH
export PKG_CONFIG_PATH=%{_libdir}/pkgconfig\${PKG_CONFIG_PATH:+:\${PKG_CONFIG_PATH}}
EOF
?
[...]
```

```
%files
%files runtime
%scl_files
?
[...]
```



# How to build a collection

- Install scl-utils-build

```
yum install scl-utils-build
```

- Build meta package

```
rpmbuild -ba metapackage.spec
```

- Use scl2spec tool for the package

```
spec2scl package.spec >packagescl.spec
```

- Build the package – use mock :)

```
rpmbuild -ba packagescl.spec --define 'scl name'
```

(or install metapackage-build instead --define 'scl name')





# 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:scl enable %{scl} - << \EOF}
  {%__python3} setup.py install --skip-build --root $RPM_BUILD_ROOT
+{%?scl:EOF}
```



# Advanced Software Collections packaging



# SCL-izing dynamic languages

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



# SCL-izing daemons

- Files placed outside of /opt

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

```
/usr/lib/systemd/system/postgresql92-postgresql
```

- Daemon started in clean environment

```
source scl_source enable postgresql92
```

```
ExecStart=/usr/bin/scl enable postgresql92 --  
/opt/<ven>postgresql92/root/usr/bin/postmaster arg
```

- SELinux

```
semanage fcontext -a -e / /opt/<ven>/postgresql92/root
```



# How to extend SCL

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

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



Are some collections available yet?



# Yes!

- RHSCCL product of Red Hat for RHEL-6/7  
devtoolset-3, httpd24, mariadb55,  
maven30, mongodb24, mysql55,  
nginx14, nginx16, nodejs010,  
perl516, php54, php55, postgresql92,  
python27, python33, ror40, ruby193,  
ruby200



# Where is community?





# So the concept exists, but...

- How to get collections in Fedora/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?**



# Work is in progress

- SCL is popular, but community not as big as we would like to see yet
- SCLs adoption in Fedora in progress to adopt
- SCLo SIG established in CentOS
  - Other SIGs would like to use SCLs in CentOS
  - Infrastructure almost ready
  - Now tweaking for SCLs



# Join [www.softwarecollections.org](http://www.softwarecollections.org)

The screenshot shows the Software Collections website interface. At the top, there is a navigation bar with the logo "software COLLECTIONS" and links for "about", "guides", and "browse collections". A search bar in the top right contains the text "php54" and a search icon. Below the navigation bar, a breadcrumb trail reads "Home / Browse Collections / Directory". The main heading is "Directory". A search filter box contains the text "php54" and a "search description" button. Below this, there are four dropdown menus: "All policies" (with a sub-option "approved only"), "All repos", "Sort: download count", and "10 per page". The search results show "Php54 by Software Collections" with a five-star rating. The description below reads: "A recent stable release of PHP with PEAR 1.9.4 and a number of additional utilities. This version..."



# Questions?

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

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

Community mailing-list: [sclorg@redhat.com](mailto:sclorg@redhat.com)

These slides already available: <http://goo.gl/ujpwgJ>

Honza Horak

e-mail: [hhorak@redhat.com](mailto:hhorak@redhat.com)

[hhorak](#) @freenode

Twitter: [@HorakHonza](#)

