

Software Collections SIG

Honza Horak <hhorak@redhat.com>

CentOS Interlock, Paris, November 2016



CentOS

Software Collections (SCL) principle

Avoid conflicts by using `/opt/rh/<stack>`

```
$ python --version
Python 2.7.5
$ scl enable rh-python35 bash
$ python --version
Python 3.3.2
```

```
$ rpm -ql rh-python35-python
/opt/rh/rh-python35/root/usr/bin/python
/opt/rh/rh-python35/root/usr/bin/python3
/opt/rh/rh-python35/root/usr/bin/python3.5
```



CentOS

What to talk about today

- Communication
- SCL packaging technology
- Building process
- Container images



Communication

The background of the slide is a dark blue sky with soft, white clouds. The clouds are more visible at the bottom and right edges, while the top and left areas are a solid dark blue.

General vision of SCL

- SCL Technology mostly solves parallel install
 - Also used for providing stacks of packages
- Modularity (Fedora) may solve the stack problem
- Containers may solve parallel install problem

General vision of SCLo SIG

- Place where broad community may build and share SW packaged as Software Collections
 - Why SCLo should be picked instead of Copr?
- Rebuild of RHSCL collections
 - Idea of having CentOS SCLo SIG to be upstream of RHSCL not working
- Common set of packages used by other SIGs and that do not conflict with base system

Existing Software Collections in SIG

devassist09
devtoolset-3
devtoolset-4
devtoolset-6
git19
httpd24
mariadb55
maven30
mongodb24
mysql55
nginx14
nginx16
nodejs010
perl516
php54
php55

postgresql92
python27
python33
rh-eclipse46
rh-git29
rh-java-common
rh-mariadb100
rh-mariadb101
rh-maven33
rh-mongodb26
rh-mongodb30upg
rh-mongodb32
rh-mysql56
rh-mysql57
rh-nginx18
rh-nodejs4

rh-passenger40
rh-perl520
rh-perl524
rh-php56
rh-php70
rh-postgresql94
rh-postgresql95
rh-python34
rh-python35
rh-redis32
rh-ror41
rh-ror42
rh-ruby22
rh-ruby23
rh-thermostat16
rh-varnish4

ror40
ruby193
ruby200
thermostat1
v8314

[sclo-git25](#)
[sclo-httpd24more](#)
[sclo-php54](#)
[sclo-php55](#)
[sclo-php56](#)
[sclo-php70](#)
[sclo-ror42](#)
[sclo-subversion19](#)
[sclo-vagrant1](#)

softwarecollections.org website

- Directory of available collections
- Technology introduction
 - Packaging guide
 - AI [SCLo]: update examples
 - AI [SCLo]: how to contribute to the scl.org content
- AI [SCLo]: Containers should be integrated

Reporting bugs for SCLs

- softwarecollections.org product in <https://bugzilla.redhat.com>
- IDEA: at least one component on bugs.centos.org?
 - non-packaging issues
 - for those who fail to find bugzilla.redhat.com
- AI [SCLo]: document clearly where to report bugs

Other communication

- More articles about usage, user stories, packaging examples, extending examples
 - [seven.c.o?](#) [planet.c.o?](#) Other?
 - “easy task” candidate
- AI [SCLo]: document how more people may write tests
- AI [SCLo]: rework wiki page (once template is ready)
- AI [SCLo]: status of SCLs (automation needed)
 - [scl.org](#) connection

The background of the slide is a dark blue, almost black, sky filled with soft, white, and light blue clouds. The clouds are more concentrated at the bottom and right sides, with some wispy clouds scattered throughout the upper half. The overall effect is a moody, atmospheric background.

SCL technology

SCL packaging technology (future)

- scl-utils 2.0
 - based on environment modules
 - in Fedora
 - a bit buggy
- spec2scl improvements
- Add SCL support to pip2rpm, gem2rpm, ...

Building process

Automation: WIP

- Too many packages in SCLo
 - almost same as in base CentOS
- (Nasty) scripts to do the work, building is triggered by human
 - takes even weeks until some builds get rebuilt
- Building from released SRPMs
 - following the order bootstrapping needed
- AI [SCLo]: fully automate rebuilding
- AI [SCLo]: automate tagging
- DONE: automate testing

Automation: testing

- How to determine
 - How long the package is in -testing repo?
 - What needs to be tested (dependencies between SCLs)?
 - Which containers to rebuild after package update?
- Ideally we would be curious what changed in a repo since X
 - Also needed for announcement messages
- Would a message bus help here?
- Would that be beneficial for other SIGs as well?

Where we keep sources

- Sources go to github.com/sclorg-distgit after build
- Look-aside cache from Fedora is used
- One can forgot to push to github
 - We may check later

Testing

- Now testing candidate repo, buildlogs, mirrors
- <https://github.com/sclorg/sclo-ci-tests/>

Building setup

- CBS Targets and tags need specific setting
 - Packages in buildroot, inheritance between tags
- We keep bothering on bugs.c.o (could we change it ourselves?)
 - no permissions to change owner as reporter
- No visibility of internal processes
 - why package is not in -candidate, when it will be available on buildlogs
- Only x86_64 now
 - 32bit or arm?
- Stats for SCL packages downloads?

The background of the slide is a dark blue gradient with soft, white, and light blue clouds scattered across it, particularly concentrated towards the bottom.

Docker Images based on SCLs

SCL Docker Container Images

- Primarily application images for OpenShift and Atomic
- Source lives on github.com/sclorg
 - Including unit tests
- PR testing (ci.centos.org)
- AI [SCLo]: test containers available on Docker Hub (age)
- Will be happy to use CCCP

Container Images in downstream

- RH internal Jenkins
 - Testing PRs on RHEL platform
 - After commit: image tested and pushed to Docker Hub
- Taking upstream code on RHEL Base image update

What do YOU need from SCLo SIG?

Honza Horak <hhorak@redhat.com>
@HonzaHorak

https://hhorak.fedorapeople.org/2016/160802_application_containers_and_system_services.pdf



CentOS