redhat.

Red Hat Software Collections

Ryan Hennessy
Sr. Solutions Architect
hennessy@redhat.com

Hello Everybody...

- Solutions Architect based out of the **FAR** west suburbs of Chicago (Iowa Adjacent)
- Husband, father of 2, brewer and drinker of beer, home automator
- Been with Red Hat for over 6 years
- Gave a similar talk at Summit **4 years ago**

Agenda

- Overview of Software Collections
- Installation/Configuration of Software Collections
- Enabling Software Collections
  - Using the SCL Utility
  - Using Docker Images
- Basic building blocks of software collection

There is no such thing as Docker....

redhat.

The Situation of Today (Some What)...

- RHEL software packages are designed for stability and long life cycles
- There is a need for updated software packages that can follow defined installation/patching mechanisms
- Provide for multiple version of the same software package on a single system

What Software Collections Provides

- Allow for multiple versions of the same software to be installed on the system
- Does not override the RHEL requirements for specific version of software
- Packaged in RPM
- Installed in a standardized path
- Easy set of commands to interact and use installed software

Red Hat Software Collections and Developer Toolset

- Built with the software collection tool set
- Packages built and supported by Red Hat
- Red Hat Developer Toolset focused on system type software development and debugging
- Red Hat Software Collections provides recent versions of dynamic programming languages, database servers, and various related packages

Red Hat Software Collections Life Cycle

- Important bug and security fixes are supplied in same manner as RHEL errata
- Individual Software Collections are Supported for 3 years. [1]
- New major version is released approximately every 18 months
- New components in RHSC have backward compatibility with the components in the previous major version of RHSC
- Available on supported 64-bit versions of RHEL 6 (Limited Software Collections)

[1] https://access.redhat.com/support/policy/updates/rhscl

What am I talking about....

Collection	Release Date	Retirement Date	Release
Apache httpd 2.4 update**	Oct 2017	Oct 2020	RHSCL 3.0
Common Java Packages***	Apr 2015	May 2019	RHSCL 2.0
DevAssistant 0.9	Oct 2014	Oct 2016	RHSCL 1.2
Eclipse 4.6	Nov 2016	Nov 2018	RHSCL 2.3
Git 1.9	Oct 2014	Oct 2016	RHSCL 1.2
Git 2.9	Nov 2016	Nov 2018	RHSCL 2.3
HAProxy 1.8	May 2018	May 2021	RHSCL 3.1
MariaDB 10.0	Apr 2015	Apr 2018	RHSCL 2.0
MariaDB 10.1	May 2016	May 2019	RHSCL 2.2
MariaDB 10.2**	Oct 2017	Oct 2020	RHSCL 3.0

So Lets chat in depth about RHSCL

- What is included:
  - Perl 5.26.1, Ruby 2.5.0, MongoDB 3.6.3 ,Varnish Cache 5.2.1, PostgreSQL 10.3, HAProxy 1.8.4, PHP 7.0.27, MySQL 5.7.21, Apache httpd 2.4.27, PHP 7.1.8, nginx 1.12.1, Python 3.6.3, Maven 3.5.0, MariaDB 10.2.8, PostgreSQL 9.6.5, MongoDB 3.4.9, Node.js 8.9.4.....
- Check the release notes for updates to features and functionalities. [1]

[1] https://access.redhat.com/documentation/en-us/red_hat_software_collections/3/

Installation/Configuration Instructions

Mommy, where do software collections come from?

- Software Collections are are provided in their yum repos
  - rhel-server-rhscl-6-rpms
  - rhel-server-rhscl-7-rpms
- Some (not all) require the optional channel to be enabled as well [1]
- RHSCL 3.1 supported on RHEL 7:
  - 64 Bit Intel
  - 64 Bit ARM
  - IBM z Systems
  - IBM POWER, little endian

[1] https://access.redhat.com/documentation/en-us/red_hat_software_collections/3/html-single/3.1_release_notes/#sect-Installation-Subscribe-Optional

To be continued…

- Software Collection packages begin with “rh-”
- Software Collections can include optional packages that are not installed by default
  - Perl software collection has a CPAN libraries
  - Python software collection has additional modules
- To find these additional packages:
  - yum list available | grep rhel-server-rhscl-7-rpms

What about Satellite

One Last Step

Install the package needed to invoke software collections

```
[root@testserver bin]# rpm -ql scl-utils
/etc/bash_completion.d/scl.bash
/etc/scl/prefixes
/opt/rh
/usr/bin/scl
/usr/bin/scl_enabled
/usr/share/man/man1/scl.1.gz
[root@testserver bin]#
```

The Down and Dirty Details

Identifying the Software Collections Installed

Run the 'scl' command to list all installed software collections

```
[root@server ~]# scl --list
devtoolset-2
mysql55
perl516
php54
python33
[root@server ~]#
```

Enabling a Software Collection

After enabling the software collection you can see that the version of the python interpreter is different

```
[root@server ~]# python --version
Python 2.6.6
[root@server ~]#
[root@server ~]# scl enable python33 bash
[root@server ~]#
[root@server ~]#
[root@server ~]# python --version
Python 3.3.2
[root@server ~]#
```

Enabling a Software Collection

- Enable multiple software collections at once by providing each collection on the same enable command
- Environmental variable 'X_SCLS' can be used to determine which collections are currently enabled

```
[root@server ~]# scl enable python33 mysql55 bash
[root@server ~]# echo $X_SCLS
python33 mysql55
[root@server ~]#
```

Enabling a Software Collection Service

- Software Collections also include packages that are services (ie. databases)
- Software Collection services are enabled the same way as any other system service

```
[root@scltest ~]# scl --list
rh-mongodb36
rh-mysql57
[root@scltest ~]#
```

```
[root@scltest ~]# systemctl status rh-mysql57-mysqld
● rh-mysql57-mysqld.service - MySQL 5.7 database server
   Loaded: loaded (/usr/lib/systemd/system/rh-mysql57-mysqld.service; disabled; vendor pres
   Active: active (running) since Mon 2018-06-04 10:37:52 CDT; 2min 28s ago
  Process: 3507 ExecStartPost=/usr/bin/scl enable $RH_MYSQL57_SCLS_ENABLED -- /opt/rh/rh-my
  Process: 3473 ExecStart=/opt/rh/rh-mysql57/root/usr/libexec/mysqld-scl-helper enable $RH_
sedir=/opt/rh/rh-mysql57/root/usr --pid-file=/var/run/rh-mysql57-mysqld/mysqld.pid (code=ex
  Process: 3403 ExecStartPre=/usr/bin/scl enable $RH_MYSQL57_SCLS_ENABLED -- /opt/rh/rh-mys
  Process: 3374 ExecStartPre=/usr/bin/scl enable $RH_MYSQL57_SCLS_ENABLED -- /opt/rh/rh-mys
  Process: 3368 ExecStartPre=/usr/bin/scl enable $RH_MYSQL57_SCLS_ENABLED -- /usr/bin/scl_e
 Main PID: 3479 (mysqld)
   CGroup: /system.slice/rh-mysql57-mysqld.service
           └─3479 /opt/rh/rh-mysql57/root/usr/libexec/mysqld --daemonize --basedir=/opt/rh/
```

Running an application using Software Collections

Simple python script: (Don't worry if you don't know python)

```
def outer():
        x = 1
        print ("Pre innner call: ", x)
        def inner():
            nonlocal x
            x = 2
            print("inner:", x)
        inner()
        print("outer:", x)

if __name__ ==  "__main__":
         outer()
```

Run the Script with the Standard Python

Using the version of python installed with RHEL

```
[root@server ~]# python pythontest.py
  File "pythontest.py", line 5
    nonlocal x
             ^
SyntaxError: invalid syntax
```

Once more with Software Collections

This time run the script with the python 3.3 software collection

```
[root@server ~]# scl enable python33 "python pythontest.py"
Pre innner call:  1
inner: 2
outer: 2
```

And now you can remember Linux Containers Exist.....

Red Hat Software Collections and Containers

- Some of the Red Hat Software Collection packages are provided as container images.
- These images can be run on RHEL 7 or RHEL Atomic Host
- Images can be found on the Red Hat Container Catalog with prefix of rhscl [1]

[1] https://access.redhat.com/containers/

Software Collection Image as a base

- Use the predefined image as your base image to build from.
- Provides all the functionality of the normal software collection just inside of a container image
- You will need to invoke the ‘scl’ command as part of your CMD statement
- Benefit from having Red Hat help manage the software components in container image

```
FROM registry.access.redhat.com/rhscl/python-35-rhel7
```

Using Source to Image Tooling for RHSCL

- Leverage the similar source to image functionality that is provide OpenShift
- Build the image that includes both the Software Collection as well as the application source
- Output similar as a Dockerfile build without having to manage the Dockerfile

How this source to image is done

- Assumptions for this build that all the requirements have been met [1]
- Install the source to image binary (Provided in the software collections repo)
- Make sure you have the extra's repo enabled for docker binary install

```
[root@scltest ~]# yum install source-to-image
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package source-to-image.x86_64 0:1.0.9-1.el7 will be installed
--> Processing Dependency: docker for package: source-to-image-1.0.9-1.el7.x86_64
--> Processing Dependency: git for package: source-to-image-1.0.9-1.el7.x86_64
--> Running transaction check
---> Package docker.x86_64 2:1.13.1-63.git94f4240.el7 will be installed
--> Processing Dependency: docker-client = 2:1.13.1-63.git94f4240.el7 for package: 2:docker-1.13.1-63.git94f4240.el7.x86_64
--> Processing Dependency: docker-common = 2:1.13.1-63.git94f4240.el7 for package: 2:docker-1.13.1-63.git94f4240.el7.x86_64
```

[1] https://access.redhat.com/documentation/en-us/openshift_enterprise/3.0/html/creating_images/creating-images-s2i

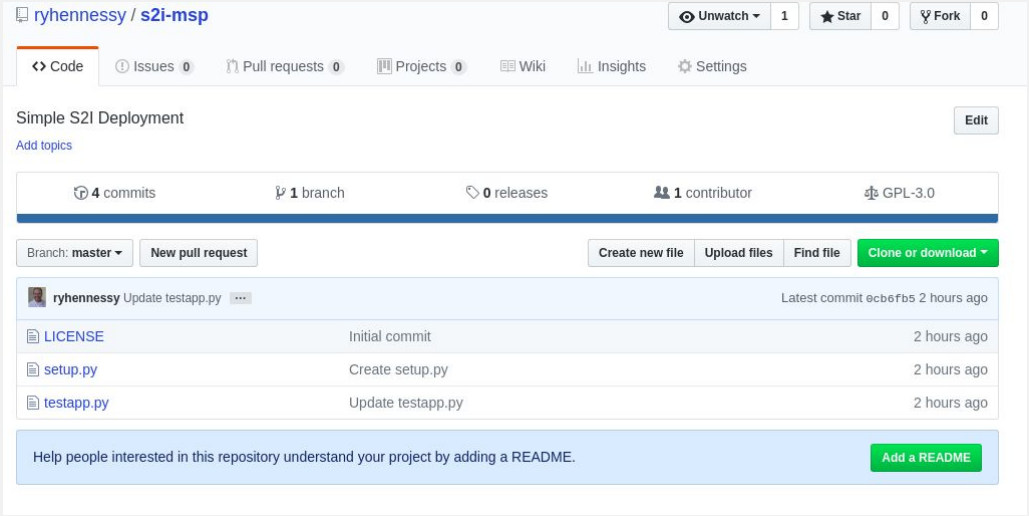
How this source to image is done

- Pull the right Docker Image from the Red Hat container registry

```
[root@scltest ~]# systemctl start docker
[root@scltest ~]# docker pull registry.access.redhat.com/rhscl/python-35-rhel7
Using default tag: latest
Trying to pull repository registry.access.redhat.com/rhscl/python-35-rhel7 ...
latest: Pulling from registry.access.redhat.com/rhscl/python-35-rhel7
b12636467c49: Extracting [======================>                            ] 34.54 MB/74.92 MB
b538cc6febe6: Download complete
fd87b2ca7715: Download complete
92783d467c55: Downloading [================================================>   ] 85.96 MB/89.85 MB
40858e6e4db1: Downloading [====================================>              ] 37.97 MB/50.7 MB
```

Put the Source Code Somewhere

- Have the source code in a git repo somewhere.
  - https://github.com/ryhennessy/s2i-msp



Run through the S2I Process

- Run the s2i build process making sure to include the git repo, the SCL image to use, as well as the output image name.
- In a interest of time we will not be able to go through the full s2i toolset.

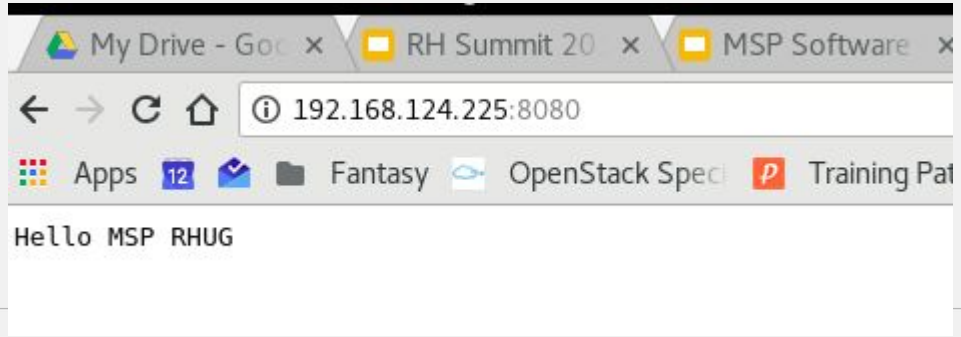
```
[root@scltest ~]# s2i build https://github.com/ryhennessy/s2i-msp.git rhscl/python-35-rhel7 my-silly-app
I0604 12:53:43.882579 01404 clone.go:32] Downloading "https://github.com/ryhennessy/s2i-msp.git" ...
I0604 12:53:44.594406 01404 install.go:251] Using "assemble" installed from "image:///usr/libexec/s2i/assemble"
I0604 12:53:44.594439 01404 install.go:251] Using "run" installed from "image:///usr/libexec/s2i/run"
I0604 12:53:44.594472 01404 install.go:251] Using "save-artifacts" installed from "image:///usr/libexec/s2i/save-artifacts"
---> Installing application source ...
---> Installing application ...
running develop
running egg_info
creating testapp.egg-info
writing requirements to testapp.egg-info/requires.txt
writing top-level names to testapp.egg-info/top_level.txt
writing testapp.egg-info/PKG-INFO
writing dependency_links to testapp.egg-info/dependency_links.txt
writing manifest file 'testapp.egg-info/SOURCES.txt'
reading manifest file 'testapp.egg-info/SOURCES.txt'
writing manifest file 'testapp.egg-info/SOURCES.txt'
running build_ext
Creating /opt/app-root/lib/python3.5/site-packages/testapp.egg-link (link to .)
Adding testapp 0.1 to easy-install.pth file
```

Lets Run the New Container Image

- Will use the newly created image “my-silly-app”
- Will expose host ports to this container image so we can view it from a web browser

```
[root@scltest ~]# docker images
REPOSITORY                                    TAG                 IMAGE ID            CREATED             SIZE
my-silly-app                                  latest              c522832801ff        2 minutes ago       628 MB
registry.access.redhat.com/rhscl/python-35-rhel7   latest         0dbd08ad57f2        13 days ago         627 MB
```

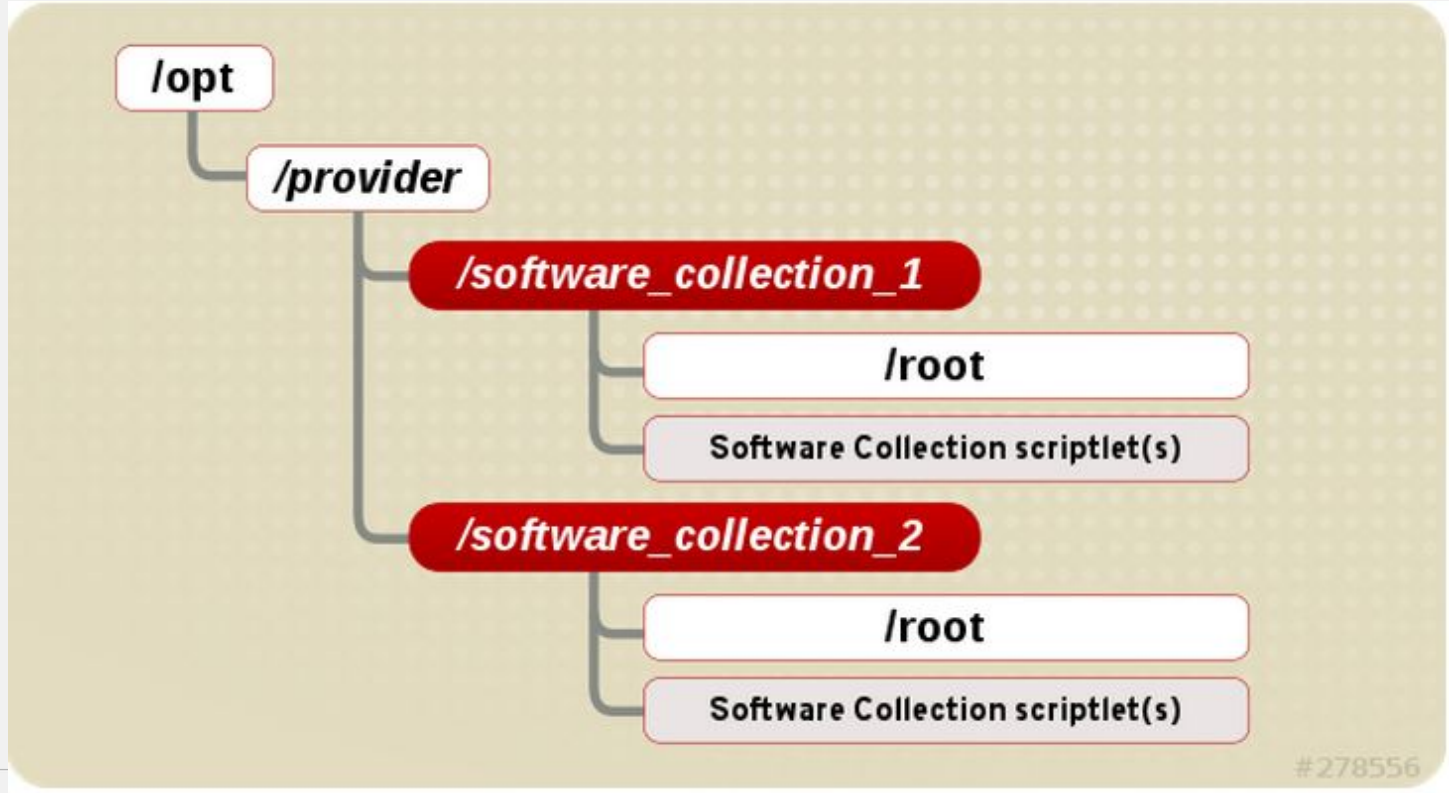
```
[root@scltest ~]# docker run -d -p 8080:8080 --name this-app-is-simple my-silly-app
f3cc68ab08e64734f94ad5f01e0b6b0f4645503ef2981a7a07915eecf802303b
```



redhat.

Lets Understand the Building Blocks of Software Collections

The Building Blocks of a Software Collection



Understanding the scl binary

/etc/scl/prefixes

Configuration directory 'scl' command uses to determine the software collection file system

```
[root@server prefixes]# pwd
/etc/scl/prefixes
[root@server prefixes]# cat python33
/opt/rh
[root@server prefixes]#
```

The Software Collections Directory Structure

/opt/<provider>/<software collection>/enable

The environmental variables that are modified when a software collection is enabled

```
[root@server python33]# pwd
/opt/rh/python33
[root@server python33]# cat enable
export PATH=/opt/rh/python33/root/usr/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/opt/rh/python33/root/usr/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
export MANPATH=/opt/rh/python33/root/usr/share/man:${MANPATH}
# For systemtap
export XDG_DATA_DIRS=/opt/rh/python33/root/usr/share${XDG_DATA_DIRS:+:${XDG_DATA_DIRS}}
# For pkg-config
export PKG_CONFIG_PATH=/opt/rh/python33/root/usr/lib64/pkgconfig${PKG_CONFIG_PATH:+:${PKG_CONFIG_PATH}}
```

The Software Collections Directory Structure

/opt/<provider>/<software collection>/root

The complete file system layout containing all the files of the software collection

```
[root@server root]# pwd
/opt/rh/python33/root
[root@server root]# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  sbin  selinux  srv  sys  tmp  usr  var
[root@server root]#
```

And in Summary....

- Software Collections tools are great for adding new software functionality/versions without steping on the system needed versions
- Red Hat's Software Collections uses:
  - Red Hat Software Collections = Updated runtimes/application
  - Red Hat Developer Tool Set = updated gcc and debugging tools
- Ease of use for existing software collections

QUESTIONS?

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos