

Package Management and Yum

Most modern Linux distributions allow administrators to add software to their systems using *packages*. Package management systems like RPM (which we'll cover a bit later in the course) provide a method for administrators and software distributors to keep track of the software that's installed and which *dependencies* must be met in order for new software to be installed. In this lesson, we'll learn about a tool called *Yum* that can be used to locate and install RPM packages and dependencies.

An Introduction to Package Management

Software packages in Linux are designed to allow software to be installed, updated, and removed efficiently. While the primary function of the software package is to contain the files that are directly associated with a piece of software, the package generally also contains information about any additional packages that must be installed in order for the software to work correctly. These additional required software packages comprise *software dependencies*. Modern package management tools have the ability to interpret these dependencies from a package and install the required packages automatically.

Package management tools that ship with Linux distributions generally have an entire infrastructure associated with their operation. The distribution you use will most likely have a set of packages maintained for it by the distributor. These packages are usually made available online in a *repository*. In order to spread the load of serving these packages up to end users, these repositories will be *mirrored* to several locations. This allows end users to download packages from the nearest and/or best performing repository. The O'Reilly School of Technology maintains its own mirror of packages for the CentOS distribution, which is the distribution used on your Linux Learning Environment machine. Others can maintain repositories of software that isn't included in the standard distribution as well, which makes it much simpler to distribute software to end users.

Package management tools download packages from the repository (both the packages you have requested to install, and the packages required to satisfy dependencies) and install them in the proper order. At the same time, they update a database on your system that contains a list of all of the packages you have installed. This database will be used for dependency checks and determining upgrades. Package management tools also give you the ability to query software repositories to determine which packages they contain. Packages in the repository can then be searched using a name or a short description of the functionality of the software, allowing administrators to locate the software they need.

The Yum Package Manager

In this course, we'll use the **yum** package management tool for downloading and installing software. This package manager is used by Red Hat Linux-based systems

(including CentOS). In order to install software using yum, you will need root privileges, which means you'll be using sudo. The general format for using yum is:

OBSERVE:

yum subcommand objects

There are quite a few yum subcommands, but you'll primarily use **search**, **install**, **update**, **upgrade**, **remove**, and **clean**. We will start with the **search** subcommand, and use it to find a piece of software called "ssh" that allows us to log into other computers on the network. Root privileges are *not* required to use **yum** to query software repositories:

INTERACTIVE SESSION:

```
[username@username-m0 etc]$ yum search ssh
Loaded plugins: fastestmirror
=====
==                               Matched:                               ssh
=====
===
libssh2.x86_64 : A library implementing the SSH2 protocol
libssh2.i686 : A library implementing the SSH2 protocol
libssh2-devel.i686 : Development files for libssh2
libssh2-devel.x86_64 : Development files for libssh2
libssh2-docs.x86_64 : Documentation for libssh2
openssh.x86_64 : An open source implementation of SSH protocol
versions 1 and 2
openssh-askpass.x86_64 : A passphrase dialog for OpenSSH and X
openssh-clients.x86_64 : An open source SSH client applications
openssh-ldap.x86_64 : A LDAP support for open source SSH server
daemon
openssh-server.x86_64 : An open source SSH server daemon
pam_ssh_agent_auth.i686 : PAM module for authentication with
ssh-agent
pam_ssh_agent_auth.x86_64 : PAM module for authentication with
ssh-agent
trilead-ssh2.noarch : SSH-2 protocol implementation in pure Java
trilead-ssh2-javadoc.noarch : Javadoc for trilead-ssh2
ksshaskpass.x86_64 : A KDE version of ssh-askpass with KWallet
support
jsch.noarch : Pure Java implementation of SSH2
python-paramiko.noarch : A SSH2 protocol library for python
python-twisted-conch.x86_64 : SSH and SFTP protocol
implementation together with clients and servers
kdeutils.x86_64 : KDE Utilities
krb5-appl-clients.x86_64 : Kerberos-aware telnet, ftp, rcp, rsh
and rlogin clients
```

```
krb5-appl-servers.x86_64 : Kerberos-aware telnet, ftp, rcp, rsh
and rlogin servers
perl-Digest-BubbleBabble.noarch      :      Create      bubble-babble
fingerprints
pexpect.noarch : Pure Python Expect-like module
```

Some items match based on package name, some match in their description, and some don't seem to match at all. The package we're interested in shows up on the list as **openssh-clients.x86_64**. Sometimes the brief description of the package found in the "yum search" output isn't enough. If you want more information, including a longer description, the size of the package, and the version, you can use the **info** subcommand. Before we install the openssh-clients package, let's take a look at its extended information. We don't need to use the full name of the package to do this. We can omit the *architecture* or *arch* string (x86_64) at the end of the package name:

INTERACTIVE SESSION:

```
[username@username-m0 ~]$ yum info openssh-clients
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Available Packages
Name           : openssh-clients
Arch           : x86_64
Version        : 5.3p1
Release        : 70.el6_2.2
Size           : 357 k
Repo           : updates
Summary        : An open source SSH client applications
URL            : http://www.openssh.com/portable.html
License        : BSD
Description    : OpenSSH is a free version of SSH (Secure SHell), a
program for logging
                : into and executing commands on a remote machine.
This package includes
                : the clients necessary to make encrypted
connections to SSH servers.
```

Installing this package will put the ssh client software on our machine. Let's go ahead and do that using the **install** subcommand. Just like the **info** subcommand, you don't need to use the full package name "openssh-clients.x86_64." You'll rarely need to specify the architecture of the package that you want to install; yum will install the correct package for your architecture in most cases:

INTERACTIVE SESSION:

```
[username@username-m0 etc]$ sudo yum install openssh-clients
...
Dependencies Resolved
```

```
=====
=====
=====
```

```
Package                               Arch
Version                               Repository
Size
```

Installing:

```
openssh-clients                       x86_64
5.3p1-70.el6_2.2                       updates
357 k
```

Installing for dependencies:

```
fipscheck                             x86_64
1.2.0-7.el6                           base
14 k
fipscheck-lib                         x86_64
1.2.0-7.el6                           base
8.3 k
openssh                               x86_64
5.3p1-70.el6_2.2                       updates
235 k
```

Transaction Summary

```
=====
=====
=====
```

```
Install      4 Package(s)
Upgrade      0 Package(s)
```

```
Total download size: 614 k
Installed size: 1.7 M
```

| Is this ok [y/N]: y

Downloading Packages:

```
-----
-----
-----
```

Total

5.1 MB/s | 614 kB 00:00

Running rpm_check_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : fipscheck-lib-1.2.0-7.el6.x86_64
1/4

```
Installing      : fipscheck-1.2.0-7.el6.x86_64
2/4
Installing      : openssh-5.3p1-70.el6_2.2.x86_64
3/4
Installing      : openssh-clients-5.3p1-70.el6_2.2.x86_64
4/4
```

```
Installed:
  openssh-clients.x86_64 0:5.3p1-70.el6_2.2
```

```
Dependency Installed:
  fipscheck.x86_64 0:1.2.0-7.el6                fipscheck-
  lib.x86_64 0:1.2.0-7.el6                      openssh.x86_64 0:5.3p1-
  70.el6_2.2
```

```
Complete!
```

Some of the output from that command was removed to save space which is indicated by the ellipsis. As our example shows, installing "openssh-clients" requires the installation of three additional packages. These packages are installed in order to resolve dependencies associated with the openssh-clients package. Let's do a quick check to make sure that ssh is working properly. In this example, we'll use ssh to log into the cold.useractive.com server. Your username and password will be the same as the one you use for your Linux Learning Environment machine. Once you've confirmed that you can log in, you can log out at any time using **exit**:

INTERACTIVE SESSION:

```
| [username@username-m0 etc]$ ssh cold.useractive.com
username@cold.useractive.com's password:
Last login: Tue Feb 28 13:10:28 2012 from
clustergw.useractive.com
| cold1:~$ exit
logout
Connection to cold.useractive.com closed.
```

Pretty neat, huh? We'll discuss ssh in depth in the next lesson. For now though, we'll continue to focus on the package manager. Let's try removing a package using (you guessed it!) the **remove** subcommand. When your system was originally installed, we included a completely unnecessary package just so we could demonstrate package removal now. (We are crafty that way sometimes.) This package is called **cdrdao**. Normally you'd install this package so that you could burn optical discs on your machine, but since you don't have a CD or DVD burner installed in your virtual machine, it's totally useless. Go ahead and remove **cdrdao** with **yum** as shown:

INTERACTIVE SESSION:

```
| [username@username-m0 ~]$ sudo yum remove cdrdao
```

Loaded plugins: fastestmirror
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package cdrdao.x86_64 0:1.2.3-4.el6 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

```
=====
=====
=====
```

Package	Arch	Repository
---------	------	------------

```
=====
=====
=====
```

Removing:
 cdrdao x86_64
 1.2.3-4.el6 @base
 1.1 M

Transaction Summary

```
=====
=====
=====
```

Remove 1 Package(s)

Installed size: 1.1 M

| Is this ok [y/N]: y

Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
 Erasing : cdrdao-1.2.3-4.el6.x86_64
1/1

Removed:
 cdrdao.x86_64 0:1.2.3-4.el6

Complete!

This package has now been removed from your system. If you remove a package that has configuration files associated with it, those files might be left on your system. By leaving those files in the system, the administrator won't lose a configuration file that

may needed later. Even so, it's always a good idea to back up any configuration files you think you may need before you do a package removal. Or, you may want to purge configuration files for removed software manually to save space or to make sure that sensitive data isn't left lying around. Yum *never* removes user data, so you will not need to worry about inadvertently removing user-generated data during a package removal.

Source: http://courses.oreillyschool.com/sysadmin2/package_management_and_yum.html