Understanding RPM Package Management Tutorial

Author Name: Chris Negus
08/31/2012

OVERVIEW

Red Hat Enterprise Linux, Fedora, and many other Linux distributions group their software together in packages using what is referred to as RPM Package Manager (RPM). The "R" in RPM originally stood for "Red Hat" but changed a few years ago to the recursive "RPM" to reflect the fact that other Linux distributions besides Red Hat use this packaging system. If you are new to administering systems that use RPM packaging, it is important that you understand what RPM packages are and how you work with them.

Using this tech brief as a tutorial you can learn about RPM packages and the commands for working with those packages. The tech brief covers the following topics:

- What are RPM packages?
- How do you use the rpm command to install and query RPM packages on your local system?
- How do you use the yum command to download, install and otherwise manage RPM packages from YUM software repositories?

WHAT ARE RPM PACKAGES?

When Linux was first created, most of the software used in Linux systems was passed around in *tarballs*. A tarball is a single archive file (created using the **`tar`** command) that can contain multiple files. So, a software project that created a Web server or a word processor application might gather up the files needed to make their application work, then distribute a tarball of those files to those who wanted to install the application. A user would untar the tarball and begin using the application. This approach had several disadvantages:

- **Hard to manage.** Once you installed the tarball, there was no way to manage the software. It would be hard to know which version of the software you had. Because files could be spread across your file system, it could be difficult to remove the software or upgrade it.
- **Dependencies.** If an application depended on other software being installed, it could be a manual process to be sure that dependent software was in place and to keep that software up to date as well.

RPM packaging set out to solve the software management problem by packaging *metadata* along with the software for an application. That metadata includes version numbers, the list of files in the package, a description of the package, information about the packager, and many other items. The metadata can also include dependencies (other software the package needs to work) and scripts (to run commands to do such things as create directories, add a user account, or turn on a service).

At first, the **`rpm`** command was the primary tool created to install and manage RPM packages. Now the **`yum`** command and related yum repositories and software channels have supplanted most (but not all) direct uses of the **`rpm`** command. On top of **`yum`**, there are now graphical tools (such as PackageKit) and automated update tools for further simplifying RPM package installation and management.

If you currently have a Red Hat Enterprise Linux installed, the best way to learn about RPM packages is to download one and investigate it yourself. The following sections help you do that.

INVESTIGATING RPM PACKAGES

You can learn something about RPM packages by simply getting an RPM package and investigating it. Assuming you have a Red Hat Enterprise Linux system to work from, here are two ways you can get an RPM package to begin following along:

- **Download a package**: If you have a Red Hat Enterprise Linux system available that is registered with Red Hat Network, you can use the **`yumdownloader`** command to get a package. For example, type the following as root from a shell.

```
# yumdownloader amanda-2*
```

- **Get a RHEL installation DVD**: If you have a Red Hat Enterprise Linux installation DVD, insert it and wait for it to automatically mount. Then open a shell and change to the Packages directory on the DVD (for example, **`cd /media/RHEL*/Packages`**).

For RHEL 6.3, the package name is **amanda-2.6.1p2-7.el6.x86_64.rpm** (the package you get may be different, as later versions are released or if you have a different type of computer). Just from the name, you can tell a lot about the package. Figure 1 illustrates the different parts of the amanda RPM package:

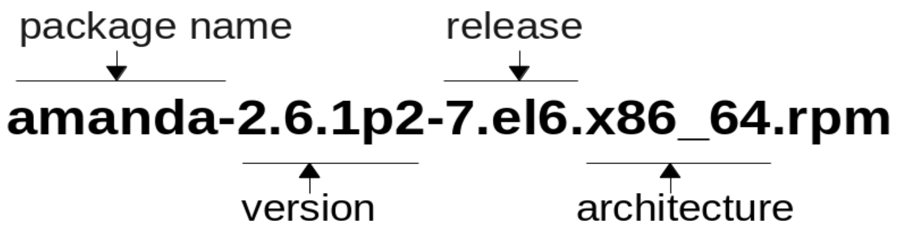

Figure 1: Breakdown of RPM package naming

The basename of the package just shown is amanda. After that, the version number (2.6.1p2) is the version assigned by the Amanda project (http://amanda.org), which tells us that the version number is 2.6.1, patch level 2. The release information (7.el6) is added by the packager (in this case, Red Hat). The release shows that this version of the package has been released 7 times (probably for bug fixes) and is associated with Red Hat Enterprise Linux 6 (el6). This particular package was built for 64-bit PC-type architecture (x86_64).

USING THE RPM COMMAND

With the package in your current directory, you can investigate it in various ways before you install it. Although the **`yum`** command is generally preferred for installing and removing packages (more on that later), the **`rpm`** command is still the best choice for:

- Querying a package (if the package is on your local file system or after the package is installed)
- Validating a package (checking a package has not been tampered with, before or after installation).

To query package, use the **`-q`** option to the **`rpm`** command, along with an indication of what you want to query. To query a package before it is installed, add the **`-p`** option along with the package name as an argument.

To see information about the amanda package in your current directory, add the **`-i`** option to the **`rpm -qp`** command line as follows:

```
# rpm -qpi amanda-2.6.1p2-7.el6.x86_64.rpm

Name     : amanda               Relocations: (not relocatable)
Version  : 2.6.1p2              Vendor: Red Hat, Inc.
Release  : 7.el6                Build Date: Wed 23 Jun 2010 07:39:33 AM EDT
Install Date: (not installed)   Build Host: x86-007.build.bos.redhat.com
Group    : Applications/System  Source RPM: amanda-2.6.1p2-7.el6.src.rpm
Size     : 1816241              License: BSD and LGPLv2 and GPLv3+ and GPLv2
Signature: RSA/8, Mon 16 Aug 2010 11:21:56 AM EDT, Key ID 199e2f91fd431d51
Packager : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL      : http://www.amanda.org
Summary  : A network-capable tape backup solution
Description :
AMANDA, the Advanced Maryland Automatic Network Disk Archiver, is a backup system
that allows the administrator of a LAN to set up a single master backup server to
back up multiple hosts to one or more tape drives or disk files.  AMANDA uses
native dump and/or GNU tar facilities and can back up a large number of
workstations running multiple versions of Unix.  Newer versions of AMANDA
(including this version) can use SAMBA to back up Microsoft(TM) Windows95/NT
hosts.

The amanda package contains the core AMANDA programs and will need to be installed
on both AMANDA clients and AMANDA servers.  Note that you will have to install the
amanda-client and/or amanda-server packages as well.
```

You can read the Summary and Description to see what the package is used for. The URL identifies the web site for the project (http://www.amanda.org). You can also see who created the RPM from this software (Red Hat), as well as when and where the software was built. The Signature line shows that the package was signed. This means that, if you have the public key associated with the private key used to sign the RPM, you can check whether or not the package has been tampered with.

Now that you know basically what the package is for, use the following command to see what files the package contains. First try the **`-l`** option to list the files the package contains:

```
# rpm -qpl amanda-2.6.1p2-7.el6.x86_64.rpm
/etc/amanda
/etc/amanda/DailySet1
/etc/xinetd.d/amanda
/usr/lib64/amanda
/usr/lib64/amanda/amanda-sh-lib.sh
/usr/lib64/amanda/chg-glue
...
```

The listing of files is shortened to save space. The full output shows that the package contains configuration files, scripts use with amanda, and documentation. You can list some of that information separately. Here's how to list configuration files in the amanda package:

```
# rpm -qpc amanda-2.6.1p2-7.el6.x86_64.rpm
/etc/xinetd.d/amanda
/var/lib/amanda/.amandahosts
```

To see documentation in the amanda package, type the following:

```
# rpm -qpd amanda-2.6.1p2-7.el6.x86_64.rpm
/usr/share/doc/amanda-2.6.1p2/COPYRIGHT
/usr/share/doc/amanda-2.6.1p2/NEWS |
/usr/share/doc/amanda-2.6.1p2/README
/usr/share/doc/amanda-2.6.1p2/README-rpm
/usr/share/man/man5/amanda-archive-format.5.gz
/usr/share/man/man5/amanda.conf.5.gz
/usr/share/man/man7/amanda-auth.7.gz
/usr/share/man/man7/amanda-scripts.7.gz
/usr/share/man/man8/amarchiver.8.gz
/usr/share/man/man8/amrestore.8.gz
```

To see changes and bug fixes associated with the package, type the following:

```
# rpm -qp --changelog amanda-2.6.1p2-7.el6.x86_64.rpm
* Wed Jun 23 2010 Jan Görig <jgorig@redhat.com> 2.6.1p2-7
- added amoldrecover description to amrecover man page
- Resolves: #593775

* Fri May 28 2010 Jan Görig <jgorig@redhat.com> 2.6.1p2-6
- removed non existing -k option from amfetchdump help
- Resolves: #596050
```

One thing you can do with the **rpm** command is check that the package itself is valid and not corrupted. Because amanda is part of the base Red Hat Enterprise Linux system, the package is signed and can be checked against a public key that should already be imported on your system. (For packages that you get from other organizations, you might need to import additional keys to check the validity of packages.)

The following command shows how to validate the amanda package (assuming the public key for the key used to sign the package has been imported and the amanda package is in your current directory):

```
# rpm -qp --checksig amanda-2.6.1p2-7.el6.x86_64.rpm
amanda-2.6.1p2-7.el6.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

From the **rpm** output, you can see the package listed as `OK`. I modified the package file (without modifying the payload) and ran the command again. As you can see below, the package is `NOT OK` as it now stands:

```
# rpm -qp --checksig amanda-2.6.1p2-7.el6.x86_64.rpm
amanda-2.6.1p2-7.el6.x86_64.rpm: rsa sha1 (MD5) PGP MD5 NOT OK
```

In most cases, you should install packages using the **yum** command (as described later). However, in cases where you have an RPM package available on your local system and there are not any components missing on the system the package needs to work, you can use **rpm** to install the package. For example:

```
# rpm -ihv amanda-2.6.1p2-7.el6.x86_64.rpm
Preparing...           ########################################### [100%]
   1:amanda            ########################################### [100%]
```

When you install an RPM package, the **rpm** command:

- Places the files the package contains into the proper locations in the file system.
- Stores the package's metadata in the local RPM database.
- In some cases, runs scripts to further configure the package.

After the package is installed, you can query information about the package from the local RPM database, instead of from the package itself. To do this, you can drop the **`-p`** option from the query commands shown earlier and use the package's basename. Here are some examples:

```
# rpm -qi amanda              Query local RPM db for amanda information
# rpm -ql amanda              Query local RPM db for all amanda files
# rpm -qc amanda              Query local RPM db for amanda config files
# rpm -qd amanda              Query local RPM db for amanda doc files
# rpm -q --changelog amanda   Query local RPM db for amanda changes
```

If you are done with the amanda package, you can remove it using the **`rpm -e`** command:

```
# rpm -e amanda
```

To learn more about RPM Package Management and the rpm command, refer to Appendix B of the Red Hat Enterprise Linux Deployment Guide:

`https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html-single/Deployment_Guide/index.html#ch-RPM`

While the **`rpm`** command is good for learning about RPM packages, the **`yum`** command (and related tools) is usually the better command for installing packages.

USING THE YUM COMMAND

The YUM facility was created because there were some things that the **`rpm`** command alone didn't do very well. In particular, although each RPM package stored a list of components it depended on, there was no way for the **`rpm`** command to satisfy those dependencies automatically. You had to hunt down each dependent package yourself and make sure you had them all available in a local directory before you could install the package you wanted. YUM changed that by:

- **Making package dependencies the responsibilities of the software packagers.** When Red Hat or another RPM-based Linux distribution created a set of RPM packages, they would store those packages in a software repository that was accessible on the network. The developers would make sure that all packages needed by other packages in the repository were in that repository as well. When someone used the **`yum`** command to ask to install a package, **`yum`** would download that package from the repository, along with any dependent packages needed to make the requested package work. Then all the necessary packages could be installed together.
- **Making package upgrades easier.** As new versions of a package are available, developers can place them in the YUM repository. PackageKit or other software services on your Red Hat Enterprise Linux system can be configured to check to see if new versions of packages you have installed are available.

When you register your system with Red Hat Network, YUM repositories associated with the base Red Hat Enterprise Linux system are enabled, along those associated with any additional software channels you have added. Once that is done, you simply use the **`yum`** command to add packages, remove packages, or query the repositories in various ways.

As an example of using the **`yum`** command, try installing the amanda-client package, as shown in the following example:

```
# yum install amanda-client
Updating certificate-based repositories.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package amanda-client.x86_64 0:2.6.1p2-7.el6 will be installed
--> Processing Dependency: amanda = 2.6.1p2-7.el6 for package: amanda-client-
    2.6.1p2-7.el6.x86_64
--> Processing Dependency: perl(Amanda::Util) for package: amanda-client
    2.6.1p2-7.el6.x86_64
      ...
--> Running transaction check
---> Package amanda.x86_64 0:2.6.1p2-7.el6 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
=============================================================================
 Package          Arch      Version       Repository                Size
=============================================================================
Installing:
 amanda-client    x86_64    2.6.1p2-7.el6   rhel-x86_64-server-6     203 k
Installing for dependencies:
 amanda           x86_64    2.6.1p2-7.el6   rhel-x86_64-server-6     546 k

Transaction Summary
=============================================================================
Install       2 Package(s)
Total download size: 749 k
Installed size: 0
Is this ok [y/N]: y
```

The **`yum install amanda-client`** command causes the **yum** command to search the YUM repository (provided by Red Hat Network) to download and install the amanda-client package. Because amanda-client requires components from the amanda package, that package is shown as required as well. Type **y** when prompted and both software packages are downloaded and installed.

Here are some other **yum** command lines that might be useful to you:

```
# yum info amanda-client         View information about a package
# yum repolist                   See a list of all enabled YUM repositories
# yum list available             List packages available from all repositories
# yum list installed             List all installed packages
# yum check-update               Check for available updated packages
# yum update                     Update all packages with available updates
# yum clean all                  Clean out all cached yum data
```

There are other **yum** options available as well. See the **yum** man page (type **`man yum`**) for more information.

To learn other features of YUM, refer to Chapter 5 of the Red Hat Enterprise Linux Deployment Guide:

https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html-single/Deployment_Guide/index.html#ch-yum