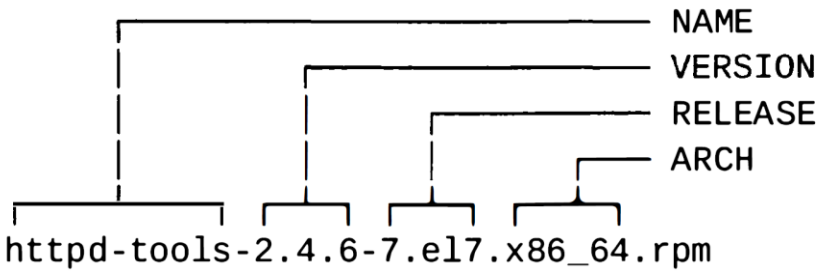
Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Red hat Package Manager:-

1. Red hat Package manager provides a standard way to package software for distribution.
2. Rpm is the default, open source and most popular package management utility for red hat systems like RHEL, CentOS and Fedora.
3. An rpm package file is installable software unit in Linux.All software provided by Red Hat for RHEL is provided as an RPM Package.
4. Rpm command allows system administrators to install, update, remove, query, verify and manage system software packages in Linux.
5. The rpm package, known as .rpm file, that includes compiled software program and libraries needed by the package. Rpm command only works with packages that build on .rpm format. Rpm keeps information installed packages under /var/lib/rpm directory, known as rpm database. The information about installed packages is stored in local RPM database on each system.
6. Rpm is the only way to install packages under Linux systems. The rpm command deals with .rpm files which contain the actual information about packages such as package name, dependencies, version etc.,
7. RPM Package files are named using a combination of the package ***name-version-release.architecture.***



- NAME is one or more words describing the contents (httpd-toos).
- VERSION is the version number of the original software (2.4.6).
- RELEASE is the release number of the package based on that version, and is packager, who might not be the original software developer (7.el7)
- ARCH is the processor architecture the package was compiled to run on. “noarch” indicates that, this package’s contents are not architecture-specific (x86_64).

8. An rpm package is a special archive made up of three components:
    - The files installed by the package
    - Information about the package (metadata); summary and description etc.,
    - Scripts which may run when this package is installed, updated or removed or which triggers when other packages are installed, updated, or removed.
9. RPMs details are stored as rpm database in /var/lib/rpm directory. **/var/lib/rpm** directory stores all files used by rpm command.
    - If you get rpm database corruption error, try to fix database problem with following two commands:

```
# rpm --rebuilddb
# rpmdb_verify Packages
```

Rpm command: -

RPM based distributions (RHEL, Fedora, Redhat, CentOS, Suse Linux) user use the ***rpm command*** to get to manage rpm packages such as install, upgrade, remove, query, verify etc.,

Syntax:-

1. To query all installed software in the system:
   rpm -qa
2. To search a particular software if installed
   rpm -qa | grep packageName
3. To search for installed software
   rpm -q packagename
4. To list all the files installed as part of particular software.
   rpm -ql packagename
5. To find a package name of a given command(to which packet the command is part of)
   rpm -qf /path/to/command
6. To list all the configuration of a particular package name
   rpm -qc RpmName
7. To list information of about an installed package.
   rpm -qi packagename
8. To install an rpm in the system
   rpm -ivh /path/to/rpmfiletoinstall.rpm
9. To update the rpm
   rpm -Uvh /path/to/rpmFiletoUpgrade.rpm
10. To uninstall (erase or remove) an rpm from the system.
    rpm -ev PackageName
11. To find rpm dependencies recursively
    rpm -qpR PakageName
12. To rebuild the currpted rpm database
    rpm --rebuilddb
13. To verify the rpm database for any errors
    rpmdb_verify Packages

RPM Software Packages and Yum:-

RPM Package files are installable software and are named using a combination of the package Name-Version-Release.Architecture such as httpd-tools-2.4.6-7.el.x86_64.rpm

When installing packages from repositories, only the package name is required. The package with the higher version will be installed. If there are multiple files with the same version, the package with the higher release number will be installed.

RPM Packages are digitally signed by the organization that packaged them. All packages from a particular source are normally signed by the same GPG private key. If the package is altered or corrupted, the signature will no longer be valid. This allows the system to verify package integrity before installing them. All RPM packages released by Red Hat are digitally signed.

Updates and Patches

When the upstream source code for a software package is patched by Red Hat, a complete RPM package is generated. If a package is newly added to a system, only the latest version of that package is needed, not every version of the package since the first release. For systems that need updating, the old version of the package is actually removed and the new version is installed. Configuration files are usually retained during an upgrade, but the exact behavior for a particular package is defined when the new version of the package is created.

In most cases, only one version or release of a package may be installed at a time. Typically, the RPM installation process will not allow files to be overwritten. If a package is built so that there are no

conflicting filenames, then multiple version may be installed. This is the case for the Kernel package. Since a new kernel can only be tested by booting to that kernel, the package is specifically designed so that multiple versions may be installed at once. If the new kernel fails to boot, the old kernel is still available.

The yum package manager:-

The rpm command may be used to install, update, remove and query RPM packages. However, it does not resolve dependencies automatically and all packages must be listed. Tools such as yum is front-end application for rpm and can be used to install individual packages or package collections(sometimes called package groups).

The yum command searches numerous repositories for packages and their dependencies so they may be installed together in an effort to alleviate dependency issues. The main configuration file for yum is */etc/yum.conf* with additional repository configuration files located in the */etc/yum.repos.d* directory. Yum cache directory is /var/cache/yum/$basearch/$releasever. The default yum log file /var/log/yum.log

Repository configuration files include, at a minimum, a repo id (in square brackets), a name and the URL location of the package repository. The URL can point to a local directory (file) or remote network share (http,ftp,etc.). If the URL is pasted in a browser, the contents should display the RPM packages, possibly in one or more subdirectories, and a repodata directory with informatioin about available packages.

The yum repository is a collection of rpm package files(Linux Software). rpm package file is a red hat package manager file and enables quick and easy software installation on redhat/CentOS linux. Yum repository hold a number of rpm package files and enable, download and installation of a new software. Yum repositories can hold rpm package files locally(local disk) or remotely(ftp, http or https). Yum configuration files hold the information required to successfully find and install rpm package files.

Yum repo files are kept under /etc/yum.repos.d/ directory with filename extension .repo. The repo file format is as below:

[REPOID]
name=rhel7
baseurl=http://alclabs.in:9090
enabled=1
gpgcheck=1
gpgkey= /etc/pki/rpm-gpg/ RPM-GPG-KEY-redhat-release

The yum command to list repositories, packages, and package groups:

yum repolist
*yum list yum**
yum grouplist

Advantages of installing software from yum repositories:-

1. Easy Software management - Installing, updating and deleting packages is simple
2. Software dependency resolution - Software dependencies are automatically resolved and installed.
3. yum repository configuration files must be located in /etc/yum.repos.d directory and must have .repo extenstion.

Working with yum:-

Finding Software with yum

1. yum list : displays installed and available packages.

ex:- yum list 'http'*

2. yum search all 'web server'
3. yum info PACKAGENAME
4. yum provides PATHNAME

ex:- *yum provides /var/www/html*

5. yum proves PathName: displays packages that match the pathnames specified (which often include wildcard characters)

Installing and removing software with yum:-

1. yum install PACKAGENAME obtains and installs a software package, including any dependencies.
Ex:- yum install httpd

2. yum update PACKAGENAME obtains and installs a newer version of the software package, including any dependencies. Generally the process tries to preserve configuration files in place, but in some cases, they may be renamed if the packager thinks the old one will not work after the update. With no PACKAGENAME specified, it will install all relevant updates.
Syntax:- yum update

Since a new kernel can only be tested by booting to that kernel, the kernel package is specifically designed so that multiple versions may be installed at once. If the new kernel fails to boot, the old kernel is still available. Using *yum update kernel* will actually install the new kernel. The configuration files hold a list of packages to "always install" even if the administrator requests an update.

3. *yum list kernel* to list all installed and available kernels.
syntax:- *yum list kernel*

4. *yum remove PACKAGENAME* removes an installed software package, including any supported packages.
Syntax:- *yum remove httpd*

Installing and removing groups of software with yum:-

1. yum also has the concept of groups, which are collections of related software installed together for a particular purpose. In RHEL7, there are two kinds of groups.
a)Regular Groups are collections of packages.
b)Environment Groups are collection of other groups which include their own packages.

The packages or groups provided by a group may be
i) mandatory (must be installed if the group is installed),
ii) default (are normally installed if the group is is installed), or
iii) optional (are not installed when the group is unless asked for specifically).

2. Like yum list, the yum group list (or yum grouplist) command will show the names of installed and available groups. Some groups are normally installed through environment groups and are hidden by default. These hidden groups can also be listed with the yum group list hidden command. If the ids option is added, the group id will also be shown. Groups can be installed, updated, removed, and otherwise queried by name or ID.

Examples:-
a. *yum group list*
b. *yum group info "Indentity Management Server"*
c. *yum group info Graphical**
d. *yum group install*

e. *yum group install "Server with GUI"*
f. *yum group install ----skip-broken "Server with GUI"*

Viewing transaction history:-

1. All install and remove transactions are logged /var/log/yum.log file
2. A summary of install and remove transactions can be viewed with yum history.
3. A transaction can be reversed with history undo options:
   ex*:- yum history undo 6*

Summary of yum commands:-

1. List installed and available packages by name
   yum list [NAME-PATTERN]
2. List installed and available groups
   yum grouplist
3. Search for a package by keyword
   yum search KEYWORD
4. Show details of a package
   yum info PACKAGENAME
5. Install a package
   yum install PACKAGENAME
   yum -y install PACKAGENAME
6. Install a package Group
   yum groupinstall "GROUPNAME"
   to remove use yum groupremove "GroupName"
7. Update all packages
   yum update
8. Remove a package
   yum remove PACKAGENAME
9. Display transaction history
   yum history
10. Revert a transaction
   yum history undo TRASACTIONID
11. Find more info about group
   yum groupinfo "GROUPNAME"
   ex:-
   yum groupinfo "Compatibility Libraries"

`12. To Enable repo ID
   yum-config-manager --enable RepoID
13. To disable repo ID
   yum-config-manager --disable RepoID
14. To install only kernel
   yum install kernel -y
15 To remove kernel
   yum remove kernel
16 To clearn yum caching
   yum clean all
17 To recreate the cache, just use
   yum repolist
18. To print information of the package
   yum info PackageName
   ex: yum info httpd
19 To search particular package based on the command

yum provides /path/to/command
example: yum provides */dump

20. To re-install the package
yum reinstall PackageName

Enabling yum Software Repositories(yum-config-manager):-

1. yum repolist all
2. You can enable or disable repositories with yum-config-manager. This will change the enabled parameter in the /etc/yum.repos.d/RepoFile. You need to install yum-utils to work with yum-config-manager

Ex:- *yum-config-manager --enable rhel7-public-beta-debug-rpms*

3.Third party repositories are directories of software package files provided by a non-Red Hat source, which can be accessed by yum from website, FTP server, or local filesystem. Yum repositories are used by non-Red Hat distributors of software, or for small collection of local packages.

4.Put a file in the /etc/yum.repos.d/ directory to enable support for a new third-party repository. Repository configuration files must end in .repo. The repository definition contains the URL of the repository, a name, whether to user GPG to check the package signatures, and if so, the URL pointing to the trusted GPG key.

If the URL for a yum repository is known, a configuration file can be created with yum-config-manager.

yum-config-manager --add -rep="http://sanjeevi.net/rheldvd71"

5.A file was created in the /etc/yum.repos.d directory with the output shown. This file can now be modified to provide a customized name and the location of the GPG key. Administrators should download the key to a local file rather than allowing yum to retrieve the key from an external source.

Creating new repository or adding rpm's to existing repository:-

You can create a repository with RPMs or add rpms to existing repository. Make sure you have installed repocreate package to do this. Here are the steps:

1. Copy rpm's to the repository directory and use the below command.

createrepo --update /RepoDir

This is create or update repomd.xml file in repodata dirctory.

2. Use yum clean all and then recreate the repo database using yum repolist command.