Package Management

RPM

- RPM – Red Hat Package Manager
- The files used by this program have an extension of .rpm
- RPM was originally created in 1997
- RPM is free and released under GPL
- It is a default package tool for RHEL, CentOS, Fedora, etc

RPM Features

- Crypto – The package in rpm can be verified cryptographically by md5 and GPG key
- Authentication – Source archive is also available which helps in authentication
- Patches – Patches can be applied which helps in updating process faster and easier
- Automated – The process is automated and non interactive
- Installation time verification – Done for dependency

Basic Tasks for RPM

- The primary features of this package management tool are

- Installing – To install a particular package
- Updating – Update the existing package.
- Uninstalling – Remove the currently installed package
- Query – Get information about the package
- Authentication – Verify the package for security reasons

Finding RPM Packages

- Red Hat developed RPM package can be found at
  - Red Hat Enterprise Linux CD RoMs
  - Red Hat Network
  - Red Hat errata page having list of packages

  It can be found on the internet. Some of the websites are:
  - http://rpmfind.net
  -http://www.redhat.com
  - http://rpm.phone.net

Installing

- Login as root or get elevated permission for the user
- Options
- -i : install a package
- -v : verbose for a nicer display
- -h : print hash marks as package archive is unpacked
- Syntax : rpm –i<option> <Package_name>

Check RPM Signatures

- Check the PGP signatures before installing any packages.
- If integrity and origin is OK then one can go ahead and install the package.
- Syntax : rpm –checksig <Package_name>

Check Dependency of RPM Package

- Check dependency of the package
- Options
- - q : Query a package
- - p : List capabilities the package provides
- -R : List capabilities on which this package depends
- Syntax : rpm –qpR <package_name>
- To ignore these dependencies use ‘—nodeps’ before installing package

Check An Installed package

- One can check if a particular package is already installed or not
- To view files of this installed package add –l option
- Syntax : rpm –ql <package_name>

View Installed RPM Packages

- One can list all recently installed packages
- One can shorten the list to check for recently installed ones by adding –last
- Syntax : rpm –qa --last

Upgrade a RPM Package

- One can upgrade a rpm package based on requirements:
- Syntax : rpm –Uvh <package_name>

Remove RPM Package

- To remove a rpm package use –e option
- In case you don’t want to remove the dependent packages use ‘—nodeps’ ption.

- Syntax : rpm –e –nodeps <package_name>

Query RPM Package

- To find the package to which particular file belongs to use –qf
- Syntax : rpm –qf <package_name>

- To find details about a particular installed package
- Syntax : rpm –qi <package_name>

Verify RPM package

- To verify a package use ‘-Vp’ option
- Syntax : rpm –Vp <Package_name>

- To verify all rpm packages, use the following command
- Syntax : rpm –Va <Package_name>

YUM - Yellow-Dog Updater Modified

- The Yum Package Management Tool
- Front-end to rpm, replacing up2date
- Configuration in /etc/yum.conf and / etc/yum.repos.d/
- Used to install, remove and list software
- yum install packagename
- yum remove packagename
- yum update packagename
- yum list avaliable
- yum list installed

DNF - Dandified YUM

- DNF is a software package manager that installs, updates, and removes packages on Fedora.
- It is the successor to YUM (Yellow-Dog Updater Modified).
- DNF makes it easy to maintain packages by automatically checking for dependencies and determines the actions required to install packages.
- eliminates the need to manually install or update the package, and its dependencies, using the rpm command.

- DNF is now the default software package management tool in Fedora.
- DNF can be used exactly as yum to search, install or remove packages.
- To search the repositories for a package type:
- # dnf search <package_name>
- To install the package:
- # dnf install <package_name>
- To remove a package:
- # dnf remove <package_name>

- Other common DNF commands include:
- autoremove - removes packages installed as dependencies that are no longer required by currently installed programs.
- Check-update - checks for updates, but does not download or install the packages.
- downgrade - reverts to the previous version of a package.
- Info - provides basic information about the package including name, version, release, and description.
- reinstall - reinstalls the currently installed package.
- upgrade - checks the repositories for newer packages and updates them.
- exclude - exclude a package from the transaction.

Automatic Updates & System Upgrades

- The dnf-automatic package is a component that allows automatic download and installation of updates. It can automatically monitor and report, via e-mail, the availability of updates or send a log about downloaded packages and installed updates.
- The Fedora system can be upgraded directly with DNF, or with the DNF system upgrade plugin.

Plugins

- The core DNF functionality can be extended with plugins. There are officially supported core DNF plugins.
- To install them run
- # dnf install dnf-plugins-core-PLUGIN_NAME

APT -GET

- Advanced package tool, or APT, is a free-software user interface that works with core libraries to handle the installation and removal of software on Debian, and Debian-based Linux distributions.
- /etc/apt contains the APT configuration folders and files.
- apt-config is the APT Configuration Query program.
- apt-config dump shows the configuration

- Syntax:
  - **apt-get [options] command**
  - **or**
  - **apt-get [options] install|remove pkg1 [pkg2 ...]**
  - **or**
  - **apt-get [options] source pkg1 [pkg2 ...]**

- apt-get install packagename
- apt-get remove packagename
- apt-get update packagename
- apt-get list avaliable
- apt-get list installed
- apt-get update
- apt-get upgrade
- apt-get dist-upgrade
- apt-get check
- apt-get clean
- apt-get -V
- apt-get --reinstall