**Package:**

* It refers to a compressed file containing all files requires to install the application and run it as well.
* These files are stored in the package according to relevant installation path on system.
* It also contains list of all dependencies.
* Examples: **.deb**: for debian based OS

.**rpm**: for red hat OS

**Dependencies:** prerequisites required to install the package

* Dependencies are list of files needed for complete installation of package and to run application successfully.
* E.g., one person develops messaging application and another one develops application to encrypt the message. Now if any person wants to write the message and encrypt it as well, messaging application needs application to encrypt the message. In this case, files needed to encrypt the message is called dependency for the application which allows to write the message.

**Package management/ Package Manager:**

* It is method of installing, updating, upgrading, querying and removing packages.
* Package managers automatically read dependencies and download them automatically (user doesn’t need to look for compatible software on websites) before proceeding with the installation of the package.
* To install package, package manager opens the archive of the package and installs files according to package specification.
* Package manager also remains aware of which files on the system belongs to which package. So at the time of uninstalling the package, package manager knows exactly files associated with relevant packages. [Unlike in Windows, it lets application installer to manage installation and uninstallation.]
* Examples: dpkg, apt-get, aptitude, synaptic for debian based OS

rpm, yum for red hat

**Repository:**

* It refers to the central place in which aggregation of data is kept and maintained in an organized way.
* Software repository contains software packages. It can also store the files on one server or across many different servers (which are called as mirrors).
* Repository is a place where package manager finds required packages.
* It is called warehouse of Linux software.

|  |  |
| --- | --- |
| **RPM(Redhat Package Manager)** | **YUM(Yellowdog Updator Modified)** |
| Back end tool / low level tool | Front end tool /high level tool |
| It doesn’t automatically check for dependencies. | It automatically check for repositories and resolve them. |
| It cannot connect to the online repositories. | It is able to connect to the online repositories. |
| It allows to install multiple versions of package simultaneously. | It notifies the user that package is already installed. It will not allow to install multiple versions of package. |
| In case, application consists of more than one command, RPM only allows to install one package at a time. | In case, application consists of more than one command, YUM allows to install multiple packages at a time in single command. |
| RPM is very useful for querying package. | YUM is useful for repository configuration. |

**Querying package using RPM:**

* list of all packages currently installed on the system,

Command: rpm -qa

* To list files comprises a particular package,

Command: rpm -ql package name

* To query a package for information,

Command: rpm -qi package name

* To determine if a particular file was put on the system as a result of installing a package,

Command: rpm -qf /path/to/file

**YUM:**

* Yum was developed for the yum dog Linux system as a replacement for the YUP (Yellow dog UPdator).
* Configuration file /etc/yum.conf provides system wide configurations.
* File /etc/yum.repos.d/ contains information regarding repositories.
* YUM contains URLs of different repositories in its configuration files.
* It allows to store configuration files of packages on more than one location on OS at a time.
* It provides CLI and GUI interface as well.
* Through network connection to local repository, YUM allows to fetch packages.

**YUM Server System Configuration**

**YUM Repository Configuration:**

* To create directory of yum repository,

Command: mkdir -p /var/ftp/pub/yum/rhel6/repodata/Packages/

* Copy all packages to above mentioned path through WinSCP.

[Make sure Linux server of VMware and local host (windows) are pingable. verify it from both end.]

* Make sure vsftpd service is running.



* Three RPM packages needs to be installed:

1. deltarpm: It contains the difference between old and new RPM package. Installing deltarpm on old RPM packages results into new RPM packages.

To install a package,

rpm -ivh deltarpm\*

1. python-deltarpm: It is needed for createrepo configuration.

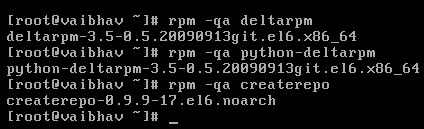
To install a package,

rpm -ivh python-deltarpm\*

1. createrepo: It is needed for custom YUM repository(local repository)(which only holds desired RPM packages) configuration.

To install a package,

rpm -ivh createrepo\*



* Createrepo command reads through custom YUM repository (in our case, ….is the location of custom YUM repository) and creates one directory called repodata. Repodata directory holds metadata for newly created repository. To add RPM packages additionally, we need to create repodata metadata with createrepo command.

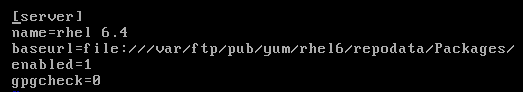
Command: createrepo <Path where packages resides>

Example**:** createrepo /var/ftp/pub/yum/rhel6/repodata/Packages/

* To use newly created repository, corresponding YUM repository configuration file (extension: **.repo**) must be created and it should be placed to /etc/yum.repos.d/ directory.

To create configuration file named server.repo,

vi /etc/yum.repos.d/server.repo

****

Repository ID: one word unique repository ID; E.g., [server]

Name: Human readable name of repository; E.g., rhel 6.4

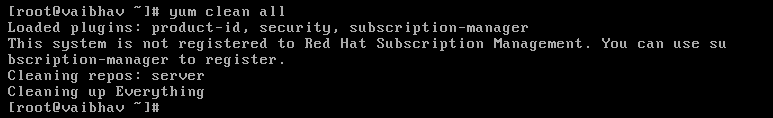
Baseurl: URL to repodata directory;

Enabled: Enable repository when performing updates and installs

Gpgcheck: GPG signature checking

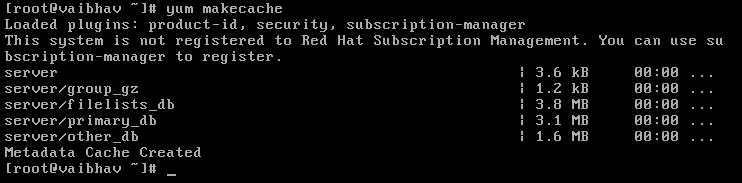
* To clean out all packages and metadata from cache,

Command: yum clean all



* To download yum repository data to cache,

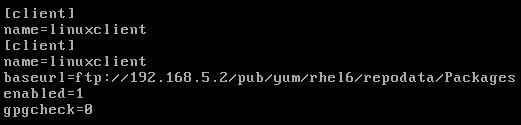
Command: yum makecache



**YUM Client System Configuration**

* Make sure server and client, both are pingable from both end.
* To create configuration file named client.repo,

vi /etc/yum.repos.d/client.repo

****

Here, 192.168.5.2 is the IP of server.

* To make sure, client system is properly configured, try to install any service on client system using yum. Like, yum install telnet

**VSFTPD: Very Secure File Transfer Protocol Daemon**

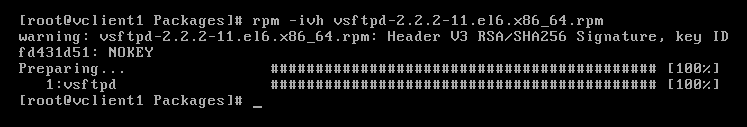
* It is a GPL (General Public License) licensed FTP server.
* It is secure, stable, extremely fast and trusted solution which supports virtual users (virtual user login does not exist as a real user login on the system in /etc/passwd & /etc/shadow file).
* It also supports IPv6 and SSL.
* To install vsftpd service,

Command: rpm -ivh vsftpd\*

i: install

v: verbose(detailed information)

h: print hash marks as package archive is unpacked.



* To check the status of vsftpd service on the system,

Command: service vsftpd status



* To start vsftpd service,

Command: service vsftpd start



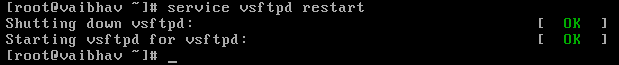
* To stop vsftpd service,

Command: service vsftpd stop



* To restart vsftpd service,

Command: service vsftpd restart



**References:**

* <http://www.computernetworkingnotes.com/network-administration/how-to-configure-yum-server-in-rhel6.html>
* <https://www.digitalocean.com/community/tutorials/how-to-set-up-and-use-yum-repositories-on-a-centos-6-vps>
* <http://www.slashroot.in/yum-repository-and-package-management-complete-tutorial>

vsftpd on client side

createrepo

update createrepo