**Install and update software packages from Red Hat Network, a remote repository, or from the local file system.**

**Presentation**

This topic is so broad that it is necessary to divide it into several parts:

* [Learn to use the rpm command](https://www.certdepot.net/sys-learn-to-use-the-rpm-command/),
* [Learn to use the yum command](https://www.certdepot.net/sys-learn-to-use-the-yum-command/),
* [Learn to deal with package groups,](https://www.certdepot.net/rhel7-get-started-package-groups/)
* [Configure a remote repository](https://www.certdepot.net/sys-configure-a-remote-repository/),
* [Configure a local repository](https://www.certdepot.net/sys-configure-a-local-repository/),
* [Manage repositories](https://www.certdepot.net/sys-manage-repositories/).

**Learn to use the rpm command.**

To install (**-i**) a new package (here **httpd-2.2.15-29.el6.x86\_64.rpm**), type:

# rpm -ivh httpd-2.2.15-29.el6.x86\_64.rpm

To upgrade (**-U**) an existing package (here **httpd-2.2.15-29.el6.x86\_64.rpm**), type:

# rpm -Uvh httpd-2.2.15-29.el6.x86\_64.rpm

To erase (**-e**) an existing package (here **httpd-2.2.15-29.el6.x86\_64**), type:

# rpm -evh httpd-2.2.15-29.el6.x86\_64

To get the list of all the installed packages, type:

# rpm -qa

To get some details about a package (here **httpd-2.2.15-29.el6.x86\_64**), type:

# rpm -qi httpd-2.2.15-29.el6.x86\_64

To get the package file list (here **httpd-2.2.15-29.el6.x86\_64**), type:

# rpm -ql httpd-2.2.15-29.el6.x86\_64

To get the list of all the modified files in the system, type:

# rpm -qVa

To get the change history of a package (here **openssl**), type:

# rpm -q --changelog openssl

\* Thu Oct 16 2014 TomÃ¡Å¡ MrÃ¡z <tmraz@redhat.com> 1.0.1e-30.4

- use FIPS approved method for computation of d in RSA

...

\* Tue Oct 26 1999 Bernhard Rosenkrdnzer <bero@redhat.de>

- inital packaging

- changes from base:

- Move /usr/local/ssl to /usr/share/ssl for FHS compliance

- handle RPM\_OPT\_FLAGS

To get the list of packages coming from a given repository (here **remi**), type:

#  rpm -qa --qf '%{VENDOR} \t %{NAME}\n' | grep -i remi

Remi Collet      libzip-last

Remi Collet      remi-release

Remi Collet      php-xml

Remi Collet      php-pecl-zip

...

**Caution:** If it is very useful to know the **rpm** command to query the system, it’s better not to use it to install a package:

* This command doesn’t deal with the package dependencies: you will have to do it yourself with all sorts of additional difficulties.
* It breaks [the yum transaction history](https://www.certdepot.net/rhel7-use-yum-transaction-history/) (=>**Warning: RPMDB altered outside of yum**).

Use the **yum** command instead.

# Learn to use the yum command.

## Presentation of yum

**yum** is a sophisticated piece of software in charge of dealing with packages and their dependencies.  
**yum** will be deprecated in future versions of **RHEL**. But its successor, **dnf**, will reuse the same syntax.

## Basic Features

To install a new package (here **httpd**), type:

# yum install httpd

To remove an existing package (here **httpd**), type:

# yum remove httpd

To get some details about a package (here **httpd**), type:

# yum info httpd

To install a package group (here **“Web Server”**), type:

# yum groupinstall "Web Server"

To remove a package group (here **“Web Server”**), type:

# yum groupremove "Web Server"

To get the list of installed package groups, type:

# yum grouplist

To get some details about a package group (here **“Web Server”**), type:

# yum groupinfo "Web Server"

To define which package provides a specified file (here **semanage**), type:

# yum whatprovides \*/semanage

To search for a particular string (here **seinfo**), type:

# yum search seinfo

To update the current base of installed packages, type:

# yum update

Note: To get the list of the packages to restart after upgrade, type: # **needs-restarting**

To get the list of all the available packages, type:

# yum list

To get the list of all the installed packages, type:

# yum list installed

To display the list of all the available but not alerady installed packages, type:

# yum list available

To get the list of the available repository, type:

# yum repolist

To clean up the yum cache, type:

# yum clean all

To get the list of the dependencies associated with a package (here **httpd)**, type:

# yum deplist httpd

**How to get started with package groups.**

**Presentation**

The package groups are a well-known feature of the **yum** command to group multiple packages under a single name.

Even though **yum** will be replaced with **dnf** in future releases of **RHEL**, the syntax and features will stay the same (at least this is the case in **Fedora 22** where you can replace **yum** with **dnf** without any problem).

Three kinds of package groups exist:

* **environment** package groups describe a **type of global configuration** containing other package groups: Minimal Install, Compute Node, Infrastructure Server, GNOME Desktop, etc.
* **top-level** package groups bring a set of package groups belonging to the **same domain**: Security Tools, Development Tools, System Administration Tools, etc.
* **simple** package groups contain packages on a **particular topic**: web-server, network-file-system-client, etc.

Also, inside a package group, there are potentially three different categories:

* **mandatory** package groups/packages are always installed.
* **default** package groups/packages are normally installed except if specified otherwise.
* **optional** package groups/packages are only installed on demand.

Note: While some **yum** subcommands **groupinfo**, **grouplist** and **groupinstall** were written with only one word in **RHEL 6**, **RHEL 7** now accepts one or two words: **yum groupinfo**/**yum group info**, **yum grouplist**/**yum group list**, **yum groupinstall**/**yum group install**, etc.

**Package Group Management**

To get the list of all the environment and top-level package groups, type:

# **yum group list ids**

Loaded plugins: fastestmirror, langpacks

There is no installed groups file.

Maybe run: yum groups mark convert (see man yum)

Loading mirror speeds from cached hostfile

...

Available environment groups:

Minimal Install (minimal)

Compute Node (compute-node-environment)

Infrastructure Server (infrastructure-server-environment)

File and Print Server (file-print-server-environment)

Basic Web Server (web-server-environment)

Virtualization Host (virtualization-host-environment)

Server with GUI (graphical-server-environment)

GNOME Desktop (gnome-desktop-environment)

KDE Plasma Workspaces (kde-desktop-environment)

Development and Creative Workstation (developer-workstation-environment)

Available Groups:

Compatibility Libraries (compat-libraries)

Console Internet Tools (console-internet)

Development Tools (development)

Graphical Administration Tools (graphical-admin-tools)

Legacy UNIX Compatibility (legacy-unix)

Scientific Support (scientific)

Security Tools (security-tools)

Smart Card Support (smart-card)

System Administration Tools (system-admin-tools)

System Management (system-management)

Done

Note1: By specifying the **ids** option, you get the system name of each package group between parenthesis. This name, called the **Group ID**, is easier to use because generally shorter and without any spaces. It is also the name used by **Kickstart** during the installation.  
Note2: To get the list of all the package groups, you need to add the **hidden** argument: **# yum group list ids hidden**  
Note3: With the **Group ID**, you don’t need to use the **group** subcommand anymore, at least for the **install** and **remove** options:

* **yum install @security-tools** = **yum group install security-tools**
* **yum remove @^web-server-environment**= **yum group remove web-server-environment**
* the **@^** prefix is reserved for **environment** groups.

To list the packages in an **environment** group (here “**Minimal Install**“), type:

# **yum group info "Minimal Install"**

Loaded plugins: fastestmirror, langpacks

Loading mirror speeds from cached hostfile

...

Environment Group: Minimal Install

 Environment-Id: minimal

 Description: Basic functionality.

 Mandatory Groups:

   +core

 Optional Groups:

   +debugging

Note1: The **core** package group contains the smallest possible installation, also called **Minimal Install**.  
Note2: The **base** package group, with the help of the **core** package group, allows to build a basic installation, also called **Infrastructure Server**.

To get the list of the packages belonging to a package group (here “**Web Server**“), type:

# **yum group info "Web Server"**

Loaded plugins: fastestmirror, langpacks

Loading mirror speeds from cached hostfile

...

Group: Web Server

 Group-Id: web-server

 Description: Allows the system to act as a web server, and run Perl and Python web applications.

 Mandatory Packages:

   +httpd

 Default Packages:

   +crypto-utils

   +httpd-manual

   +mod\_fcgid

   +mod\_ssl

 Optional Packages:

   certmonger

   libmemcached

   memcached

   mod\_auth\_kerb

   mod\_auth\_mellon

   mod\_nss

   mod\_revocator

   mod\_security

   mod\_security\_crs

   perl-CGI

   perl-CGI-Session

   python-memcached

   squid

Note: The **+** indicates which packages will be installed.

To install a package group (here “**Web Server**“), type:

# **yum group install "Web Server"**

Now, if we request some information about the **Web Server** package group, we get:

# **yum group info "Web Server"**

Loaded plugins: fastestmirror, langpacks

Loading mirror speeds from cached hostfile

...

Group: Web Server

 Group-Id: web-server

 Description: Allows the system to act as a web server, and run Perl and Python web applications.

 Mandatory Packages:

   =httpd

 Default Packages:

   =crypto-utils

   =httpd-manual

   =mod\_fcgid

   =mod\_ssl

 Optional Packages:

   certmonger

   libmemcached

   memcached

   mod\_auth\_kerb

   mod\_auth\_mellon

   mod\_nss

   mod\_revocator

   mod\_security

   mod\_security\_crs

   perl-CGI

   perl-CGI-Session

   python-memcached

   squid

Note: The **=** indicates installed packages as a part of the package group.

To install all the packages belonging to a package group (here **“Web Server”**), type:

# **yum --setopt=group\_package\_types=mandatory,default,optional groupinstall**

**"Web Server"**

**Configure a remote repository.**

To configure an access to a remote repository, create the **/etc/yum.repos.d/remote.repo** file and add the following lines:

[base]

name=CentOS-$releasever - Base

baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/

enabled=1

**Configure a local repository.**

Create a directory where packages will be stored:

# mkdir /repo

Two options: mount the distribution DVD or copy it into **/repo**.  
To mount the DVD, edit the **/etc/fstab** file and add the following line:

/dev/cdrom /repo iso9660 loop 0 0

Then, mount the DVD:

# mount -a

To copy the DVD, type:

# mount -o loop /dev/cdrom /mnt

# cd /mnt

# tar cvf - . | (cd /repo; tar xvf -)

# cd /; umount /mnt

Optionally, update the repository structure (if new packages have been added):

# yum install -y createrepo

# createrepo /repo

Create the **/etc/yum.repos.d/local.repo** file and add the following lines:

[repo]

name=Repo - Base

Baseurl=file:///repo

enabled=1

Note: If you don’t remember the syntax, read the **yum.conf** man page.

Clean up the yum cache:

# yum clean all

**Manage repositories.**

It is perfectly possible to point at several software repositories at the same time.  
To better deal with this situation, you need to use the **yum repolist** and **yum-config-manager** commands.

To get the list of all the available repositories (enabled and disabled), type:

# yum repolist all

...

repo id repo name status

base/7/x86\_64 n40l - Base enabled: 8,465

epel/x86\_64 Extra Packages for Enterprise Linux disabled

repolist: 8,465

Note: The **-v** option displays even more details.

To get the list of the enabled repositories, type:

# yum repolist

...

repo id repo name status

base/7/x86\_64 n40l - Base 8,465

repolist: 8,465

To get the list of the disabled repositories, type:

# yum repolist disabled

...

repo id repo name

epel/x86\_64 Extra Packages for Enterprise Linux 7 - x86\_64

repolist: 0

To permanently enable a repository (here **epel**), type:

# yum-config-manager --enable epel

To permanently disable a repository (here **epel**), type:

# yum-config-manager --disable epel

To temporary enable a repository (here **epel**) in order to install a specific package (here **Pound**), type:

# yum --enablerepo=epel install Pound

Note: Use the **–disablerepo** option to temporary disable a repository.