

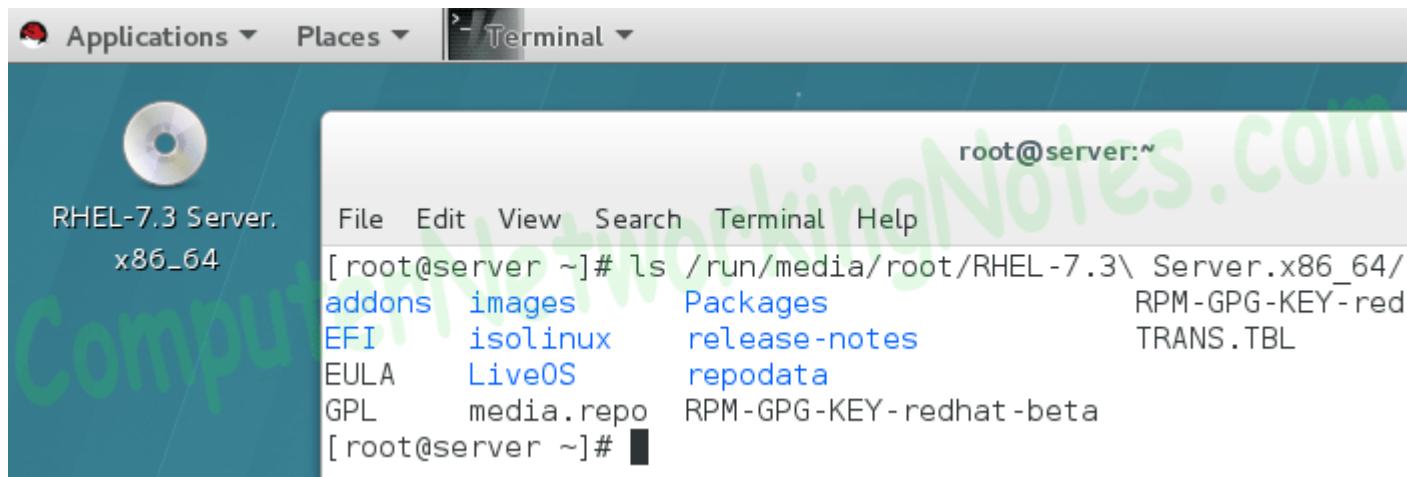
will not work with RHEL system. In Red Hat Linux repositories are provided through the RHN network. To use RHN network we need to have an active subscription. During the installation, RHEL asks us to register the system with RHN network. If system is registered with RHN network and subscription is active, default configuration is sufficient to install software packages. If system is not registered with RHN network, it will never be able to connect with Red Hat official repositories. For this reason it is recommended to use Centos instead of Red hat, if free version of RHEL Linux is needed.

## How to create a local yum repository

Before we create repository, we need to get all necessary RPM package files. For this, we can use RHEL/Centos installation disk. As we know repository is operating system specific, we have to use the same disk which we used to install the operating system. If RHEL is installed, we have to use the RHEL installation disk. If Centos is installed then we have to use the Centos installation disk.

Let's insert installation disk in system and mount it.

By default all media disks are mounted in **/run/media/UserName/** directory. Here **UserName** is the name of logged in user.



Create a directory. Since this directory will be used to store all RPM package files from disk, make sure we have sufficient (at least 4GB) free disk space. In this tutorial we will use **/rhcelab/repo** directory.

```
[root@server ~]# mkdir /rhcelab
[root@server ~]# mkdir /rhcelab/repo
[root@server ~]# df -h /rhcelab/repo/
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/rhel-root  9.8G  4.7G  5.1G  48% /
[root@server ~]#
```

Copy all RPM package files from installation disk to **/rhcelab/repo** directory. Replace **/rhcelab/repo** with your own path, if you are using other directory.

Switch to **/rhcelab/repo** directory and install **createrepo** package.

```
[root@server ~]# cd /rhcelab/repo/  
[root@server repo]# rpm -ivh createrepo-0.9.9-26.el7.noarch.rpm  
warning: createrepo-0.9.9-26.el7.noarch.rpm: Header V3 RSA/SHA256 Signature, ke  
Preparing... ###### [100%]  
Updating / installing...  
 1:createrepo-0.9.9-26.el7.noarch ###### [100%]  
[root@server repo]#
```

Now run **createrepo /rhcelab/repo** command.

```
[root@server repo]# createrepo /rhcelab/repo/  
Spawning worker 0 with 4751 pkgs  
Workers Finished  
Saving Primary metadata  
Saving file lists metadata  
Saving other metadata  
Generating sqlite DBs  
Sqlite DBs complete  
[root@server repo]#
```

During this process several cache files are generated. Let's remove them.

```
[root@server repo]# yum clean all  
Loaded plugins: langpacks, product-id, search-disabled-repos,  
                  : subscription-manager  
This system is not registered to Red Hat Subscription Management.  
  You can use subscription-manager to register.  
Cleaning repos: rhcerepo  
Cleaning up everything  
[root@server repo]#
```

That's all we need, our local repository is ready to use. But wait at this point yum command can't use this repository.

*Can you figure out why?*

Yep, you got it right. Yum command uses repository configuration files to figure out the available repositories. And so far we haven't created the necessary repository configuration file for our local repository.

Create an **rhcelp.repo** file in **/etc/repos.d/** directory with following contents

```
name=rhcerepo  
baseurl=file:///rhcelab/repo  
enabled=1  
gpgcheck=0
```

File contents

:wq → Save and exit from file

```
[root@server repo]#
```

Let's understand the meaning of above contents step by step

vi /etc/yum.repos.d/rhcelab.repo	As we know repository configuration files are stored in <b>/etc/yum.repos.d/</b> directory with an extension <b>.repo</b> , So we executed this command to create the necessary configuration file for repository.
[rhcerepo]	This is the label of repository. Usually a repository file contains configuration for multiple repositories. In that case label is used as identifier of repository.
name=rhcerepo	This configuration value is used to set the name of repository.
baseurl=file:///rhcelab/repo	This configuration value defines the location of rpm files.
enabled=1	This key defines the state of repository. If value is set to <b>1</b> then repository is enabled. If value is set to <b>0</b> then repository is disabled.
gpgcheck=0	This key defines whether the integrity of package should be check or not. If value is set to <b>1</b> , integrity will be checked. If value is set to <b>0</b> , integrity will not be checked.
:wq	We used vi editor to create the file. In vi editor, the command: wq is used to save and quit from file.

## How to define the baseurl key

To define local directory path use file://directory\_path.

To define FTP path use ftp://hostname/path\_to\_directory\_which\_has\_rpms

You may use IP address at the place of hostname, if DNS is not configured.

## The gpgcheck key

By default yum is configured to use online repositories. If repository server is compromised, a hacker may put some fake packages in repository or may change existing packages. If these packages are installed in system, the system may get hacked also. The GPG key is the counter measurement of this kind of security threat. The packages in repository are signed with GPG key. The owners of packages also provide the GPG keys. With GPG key yum can check whether the package is in its original state or has been modified in repository. Since in our local repository we put all packages from original installation disk, there is no need to enable the GPG check feature.

We can verify the newly created repository with **yum repolist** command.

```
[root@server repo]# yum repolist
Loaded plugins: langpacks, product-id, search-disabled-repos, sub
This system is not registered to Red Hat Subscription Management.
repo id          repo name          status
rhcerepo         rhcerepo          4,751
repolist: 4,751
[root@server repo]#
```

As output confirms that our repository (name – rhcerepo and id - rhcerepo) is ready to use. Currently this repository contains 4751 packages.

Now we have a working repository, in remaining part of this tutorial we will learn how to use the repository.

## Linux package management with yum command

The yum command is used to work with repository. Before we explore yum command in details let's have a quick look on some important options used with this command.

Option / Sub command	Description
provides / whatprovides	Perform a deep scan in all rpm for specified file.
search / search all	Search for specified package
list installed	List all installed packages in system

list all	List all packages available for installation from all enabled repositories as well as those that are already installed.
group list	List all groups that are available for installation
info	Provide detailed information about package
install	Install the specified package
remove	Remove the specified package
update	Update the specified package

## Searching packages

To install, remove or update a package we first need to know the exact name of the package. Since packages name contain a lot of information in name (such as version, architecture, platform etc.), they are hard to remember. We can use two sub commands to get the exact name of a package; **search** and **provides**.

To find out all packages which have specified string in name we can use following command

```
#yum search searchString
```

For example to search any package which contains **semanage** in the name we can use following command

```
#yum search semanage
```

By default this command will search for specified string only in name and summary.

To search everywhere in package we have to use **all** option with this command.

Following figure illustrates the use of both commands.

```
This system is not registered to Red Hat Subscription Management.
You can use subscription-manager to register.
rhcrepo | 2.9 kB 00:00
rhcrepo/primary_db | 3.8 MB 00:00
=====
N/S matched: semanage =====
libsemanage-python.x86_64 : semanage python bindings for
                           : libsemanage
libsemanage.i686 : SELinux binary policy manipulation library
libsemanage.x86_64 : SELinux binary policy manipulation library

Name and summary matches only, use "search all" for everything.
[root@server repo]#
```

```
[root@server repo]# yum search all semanage
Loaded plugins: langpacks, product-id, search-disabled-repos,
                 : subscription-manager
This system is not registered to Red Hat Subscription Management.
You can use subscription-manager to register.
=====
Matched: semanage =====
libsemanage-python.x86_64 : semanage python bindings for
                           : libsemanage
libsemanage.i686 : SELinux binary policy manipulation library
libsemanage.x86_64 : SELinux binary policy manipulation library
[root@server repo]#
```

If we know the command but don't the package name which provides the specific command, we can use **provides** or **whatprovides** sub command. For example to know which package provides **chocon** command we can use following command

```
#yum provides */chocon
```

This command will perform a deep scan in all RPM for file name chocon. Linux commands have an associated script file which executes when command is called at shell prompt.

We can also use **whatprovides** sub command instead of **provides** command.

```
#yum whatprovides */chocon
```

There is no functional difference between **provides** and **whatprovides** sub command.

Following figure illustrates the use of booth commands.

```
    . subscription-manager
This system is not registered to Red Hat Subscription Management.
use subscription-manager to register.
coreutils-8.22-18.el7.x86_64 : A set of basic GNU tools commonly
                               : shell scripts
Repo      : rhcerepo
Matched from:
Filename  : /usr/bin/chcon
[root@server ~]# █
[root@server ~]# yum whatprovides */chcon
Loaded plugins: langpacks, product-id, search-disabled-repos,
                 : subscription-manager
This system is not registered to Red Hat Subscription Management
use subscription-manager to register.
coreutils-8.22-18.el7.x86_64 : A set of basic GNU tools commonly
                               : shell scripts
Repo      : rhcerepo
Matched from:
Filename  : /usr/bin/chcon
[root@server ~]# █
```

Once we got the exact package name, we can use **info** sub command to get the more detailed information about the package. For instance, in above example we figured out that chcon command is provided by coreutils package. To get more information about this package we can use following command

```
[root@server ~]# yum info coreutils
Loaded plugins: langpacks, product-id, search-disabled-repos,
                 : subscription-manager
This system is not registered to Red Hat Subscription Management. You
use subscription-manager to register.
Installed Packages
Name      : coreutils
Arch     : x86_64
Version   : 8.22
Release   : 18.el7
Size      : 14 M
Repo      : installed
From repo : anaconda
Summary   : A set of basic GNU tools commonly used in shell scripts
URL      : http://www.gnu.org/software/coreutils/
License   : GPLv3+
Description: These are the GNU core utilities. This package is the
             : combination of the old GNU fileutils, sh-utils, and
             : textutils packages.

[root@server ~]# █
```

## Installing Package

package depends on other files, yum will also install those files automatically. During installation if yum finds any file which is already installed in system from package, it will check its current version. If package contains latest file, it will update the existing file. By default yum command will ask for confirmation. To perform installation without confirmation we have to use **-y** option.

For example following command will install the **vsftpd** package with confirmation

```
[root@server ~]# yum install vsftpd
Loaded plugins: langpacks, product-id, search-disabled-repos,
This system is not registered to Red Hat Subscription Management.
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.x86_64 0:3.0.2-21.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch    Version      Repository      Size
=====
Installing:
vsftpd           x86_64  3.0.2-21.el7  rhcerepo       169 k

Transaction Summary
=====
Install 1 Package

Total download size: 169 k
Installed size: 348 k
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : vsftpd-3.0.2-21.el7.x86_64
  Verifying   : vsftpd-3.0.2-21.el7.x86_64

Installed:
  vsftpd.x86_64 0:3.0.2-21.el7

Complete!
[root@server ~]#
```

Following command will install package without confirmation

```
THIS SYSTEM IS NOT REGISTERED TO RED HAT SUBSCRIPTION MANAGEMENT.  
Resolving Dependencies  
--> Running transaction check  
---> Package vsftpd.x86_64 0:3.0.2-21.el7 will be installed  
--> Finished Dependency Resolution  
  
Dependencies Resolved  
  
=====  


| Package | Arch   | Version      |
|---------|--------|--------------|
| vsftpd  | x86_64 | 3.0.2-21.el7 |

  
Installing:  
vsftpd  
  
Transaction Summary  
=====  
Install 1 Package  
  
Total download size: 169 k  
Installed size: 348 k  
Downloading packages:  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : vsftpd-3.0.2-21.el7.x86_64  
  Verifying : vsftpd-3.0.2-21.el7.x86_64  
  
Installed:  
  vsftpd.x86_64 0:3.0.2-21.el7  
  
Complete!  
[root@server ~]#
```

## Removing packages

To remove a package we can use remove sub command. Just like install, remove operation is also performed with dependency which means yum will not only remove the specified package but all remove all the packages that depend on it. For this reason, even it supports **-y** option, we should not perform remove operation without confirmation. Remove operation should always be performed with confirmation.

Following command will remove **vsftpd** package with confirmation.

```
Resolving Dependencies
--> Running transaction check
--> Package vsftpd.x86_64 0:3.0.2-21.el7 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version
=====
Removing:
vsftpd           x86_64   3.0.2-21.el7

Transaction Summary
=====
Remove 1 Package

Installed size: 348 k
Is this ok [y/N]: y  Confirmation
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Erasing    : vsftpd-3.0.2-21.el7.x86_64
  Verifying   : vsftpd-3.0.2-21.el7.x86_64

Removed:
  vsftpd.x86_64 0:3.0.2-21.el7

Complete!
[root@server ~]#
```

That's all for this tutorial. In next tutorial we will learn another Linux topic in detail.



---

Created By ComputerNetworkingNotes   Published On 2018-01-22 12:12:06   Last Updated  
2018-02-03 06:49:36

---

[ComputerNetworkingNotes](#)   [RHCE 7 Study Guide](#)   How to Configure YUM Repository in RHEL Linux

[How to Configure Samba Server in Redhat Linux](#)

[How to configure NIS Server in Linux](#)

technical issue) or simply just say to hello mail us  email us  
ComputerNetworkingNotes@gmail.com

---

Computer Networking Basic Tutorials and Study Guides © 2018. All Rights Reserved.

[Terms and Conditions](#) [Privacy Policy](#)