Linux Academy

Hands-on Lab

Red Hat/ CentOS Package Management and Repositories

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- The IP address of your server is located on the Live! Lab page
- Username: linuxacademy
- Password: 123456
- Root Password: 123456

Related Courses

LPIC-1: System Administrator - Exam 101

Related Videos

Use RPM and Yim Package Management - Working with Yum

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

In this lab, we'll learn about repositories and the `yum` package management tool on Red Hat and CentOS systems. `yum`, an acronym for "Yellow Dog Updater Modified," allows you to easily install software packages and their dependencies.

Because `yum` makes changes to the system, we'll need to run most of the commands in this lab as *root* or with the `sudo` prefix.

Configure Yum

Log in to the server using the credentials provided on the Hands-on Lab page.

The `yum` configuration file is located at `/etc/yum.conf`. To start, open it up with a text editor. We'll use `nano` since `vim` is not installed on all Red Hat and CentOS systems by default.

```
[linuxacademy@ip] sudo nano /etc/yum.conf
```

At the top of the file, we find settings that we can modify to change the way the `yum` package manager behaves. Further down, we'll find a place to insert repositories from which `yum` will download package information. However, it's more common to separate your repositories in separate files, which we'll discuss next.

Package Repositories

Repositories are the locations from which your package manager downloads software and dependencies. Although these can be set in the `yum` configuration file itself, they're more commonly found in the `/etc/yum.repos.d` directory. Let's take a look:

```
[linuxacademy@ip] cd /etc/yum.repos.d
[linuxacademy@ip] ll
total 36
-rw-r--r--. 1 root root 1991 Mar  7 04:16 CentOS-Base.repo
-rw-r--r--. 1 root root  647 May 18  2016 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  289 May 18  2016 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  630 May 18  2016 CentOS-Media.repo
-rw-r--r--. 1 root root 6259 May 18  2016 CentOS-Vault.repo
-rw-r--r--. 1 root root 2150 Mar  7 04:54 elrepo.repo
-rw-r--r--. 1 root root  957 Jun  3  2016 epel.repo
-rw-r--r--. 1 root root 1056 Jun  3  2016 epel-testing.repo
```

Each of the files in this directory ends in `.repo` and represents a repository. Filenames beginning with `CentOS` indicate repositories managed by CentOS.

For example, we can see the base CentOS repositories by viewing the contents of `CentOS-Base.repo`.

```
[linuxacademy@ip] cat CentOS-Base.repo
```

The output will include a number of entries similar to the following:

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.
org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

The bracket-enclosed heading, in this case `[base]`, indicates that these are the base repositories for the system. Under the heading, you'll find `name`, which is just a description of the repository.

The `mirrorlist` and `baseurl` items are similar to one another, but only one may be active at any time. The `baseurl` is used to download packages from one single repository location. The `mirrorlist` is a list of repository mirrors from which `yum` will choose the best candidate when installing packages.

The `gpgcheck` item is a Boolean, or binary, value. This means its default setting of `1` indicates it is active. The `gpgcheck` ensures that when you download a package, its identifier matches the one provided by the repository by way of a GPG key. The key used to perform the check is indicated by `gpgkey`.

In some repositories, we'll see an additional field not listed above:

```
enabled=0
```

This is a Boolean value that tells us this repository is *not* enabled. If a repository does not have an `enabled` setting defined, it is considered enabled by default.

View the Log File

The `yum` tool keeps a log of package transactions, which may be helpful when dealing with dependency issues or packages that don't download properly. To see this log:

```
[linuxacademy@ip] cd /var/log
[linuxacademy@ip] sudo cat yum.log
```

The output will show a list of packages, the transaction (for example, "Installed" or "Updated") and the timestamp at which each operation occurred. Since we haven't installed or updated anything yet, our log file should be empty when we view the file. We'll come back to this once we install our first package.

Note that the location of the `yum` log is defined in the `/etc/yum.conf` configuration file. `/var/log/yum.log` is just the default location.

Use the Yum Tool

The `yum` tool is fairly simple to use, and doesn't usually require flags (options indicated by a dash) to operate. This is one of the biggest visual differences between its commands and those of a lower-level package tool like `rpm`.

In this section, we'll go over a few of the most common `yum` commands.

Update Packages

The `update` command allows you to pull the latest package information from your repositories into your local cache. However, unlike its APT counterpart on Debian, `yum update` actually installs the most recent versions as well.

Let's try it out:

```
[linuxacademy@ip] sudo yum update
```

Notice that the output includes well-defined sections for each configured repository. If there are packages in need of updating, we may see a prompt asking us whether we'd like to continue. Enter `y` to proceed with the update.

There is also an `upgrade` command, which does the same thing as `update`.

Install Packages

To install a package, we'll use the `install` command. Let's try it out by installing `telnet`.

```
[linuxacademy@ip] sudo yum install telnet
```

This searches through your configure repositories and installs the latest available version of the package specified. When the search completes, your output will include a transaction summary, which you'll be prompted to accept or decline before the download takes place.

We can also install the `mysql-server` package to get an idea of the variation between different transaction summaries.

```
[linuxacademy@ip] sudo yum install mysql-server
```

When this package completes its installation, notice that the transaction summary includes information about dependency packages that were downloaded as well.

Enable a Repository

Earlier, we looked at the `enabled` setting for each repository in the configuration files. We can also enable a repository for a single transaction. This is useful if we want one package from a particular source, but don't want our entire system to use that repository.

Let's check the repositories file to find one that's disabled.

```
[linuxacademy@ip] cat /etc/yum.repos.d/CentOS-Base.repo
```

In this file, we see that the `centosplus` repository is disabled, so we'll use that for our example.

First, we'll install the `postfix` package using the installation method we learned before:

```
[linuxacademy@ip] sudo yum install postfix
```

We can see from the output that the package is downloaded from the `base` repository by default. However, if we want to get an updated version from the `centosplus` repository instead, we can enable the repo for a single transaction from the command line.

```
[linuxacademy@ip] sudo yum install --enablerepo centosplus postfix
```

This time, the output tells us that the package is being downloaded from the `centosplus` repository instead

Download RPM Files

Up to this point, we've been using `yum` to install packages automatically. However, we can also use it to download RPM files without installing them.

```
[linuxacademy@ip] sudo yum install --downloadonly telnet
```

This pulls the package in `.rpm` format without installing it on our system. However, `yum` does not allow us to specify a location for the download. Instead, we can find them in the `/var/cache/yum/x86_64/` directory. Note that parts of this file path may vary slightly across OS architectures or distribution releases. We're using the above path since we're in an x86_64 architecture, using CentOS 6.

```
[linuxacademy@ip] cd /var/cache/yum/x86_64/6/base/packages
[linuxacademy@ip] ls -al telnet*
-rw-r--r-- 1 root root 59332 Jul 10  2014 telnet-0.17-48.el6.x86_64.rpm
```

Yumdownloader

While `yum` itself doesn't allow us to download packages to a specified location, we can do this and much more with a related tool called `yumdownloader`.

Download Source Code

If we want to download source code, whether to modify it or compile it ourselves, we can use the `--source` flag.

```
[linuxacademy@ip] sudo yumdownloader --source telnet
```

This downloads an `rpm` package containing the source code if it is avaiable.

Check Source URL

Earlier, we saw that packages are often downloaded from one of a number of repository mirrors. If we want to see which mirror URL we're downloading a given package from, we can use the `--urls` flag.

```
[linuxacademy@ip] yumdownloader --urls telnet
```

Resolve Dependencies

In the event we want to download a package that has dependencies, we can use the `--resolve` flag to get them along with the package itself.

```
[linuxacademy@ip] sudo yumdownloader --resolve telnet
```

Specify a Download Location

Unlike `yum`, we can use `yumdownloader` to download a package into a directory of our choosing with the `--destdir` flag, which is short for "destination directory."

```
[linuxacademy@ip] sudo yumdownloader --destdir /opt telnet
```

We can verify by changing to the `/opt` directory and listing its contents.

```
[linuxacademy@ip] cd /opt
[linuxacademy@ip] ls -al telnet*
-rw-r--r-- 1 root root 59332 Jul 10  2014 telnet-0.17-48.el6.x86_64.rpm
```

Install and Update Without Prompting

When installing and updating packages with `yum`, we're prompted to enter `y` when If we want to update or install a package quickly without being prompted, we can use the `-y` option when running the command.

```
[linuxacademy@ip] sudo yum install -y mysql-server
```

This may be useful when scripting automatic updates or installation for a build system.

Review

The YUM package manager is a simple but powerful tool that we can use to install packages, update our system, and much more. Once you master it, you'll be prepared to not only pass your LPIC exam, but to manage software on any Red Hat or CentOS system with ease.