Linux Academy

Hands-on Lab

Use Debian Package Management Repositories and the APT Tool

Contents

Lab Connection Information

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

Related Courses

LPIC-1 Linux Administrator - Exam 101

Related Videos

Using Debian Package Management (Repositories and Package Managers)

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 `apt` package management tools on Debian and Ubuntu systems. `apt`, an acronym for "Advanced Package Tooli," allows you to easily install software packages and their dependencies.

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

Package Repositories

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

Repositories are the locations from which your package manager downloads software and dependencies. To see these locations, navigate to the `/etc/apt` directory.

```
[linuxacademy@ip] cd /etc/apt
[linuxacademy@ip] ls -al
total 40
drwxr-xr-x  6 root root  4096 Feb 21 23:49 .
drwxr-xr-x 91 root root  4096 Apr 21 21:00 ..
drwxr-xr-x  2 root root  4096 Apr 21 21:00 apt.conf.d
drwxr-xr-x  2 root root  4096 Apr 14  2016 preferences.d
-rw-r--r--  1 root root  3335 Apr 21 20:58 sources.list
drwxr-xr-x  2 root root  4096 Apr 14  2016 sources.list.d
-rw-r--r--  1 root root 12255 Feb 21 23:46 trusted.gpg
drwxr-xr-x  2 root root  4096 Apr 14  2016 trusted.gpg.d
```

In this directory, we'll find several important files and subdirectories. Let's start by looking at `sources.list`, which contains the repositories that came installed on the system. Open the `sources.list` file in your text editor of choice. You'll see a number of lines that look like this:

```
deb http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ xenial main
restricted
deb-src http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ xenial main
restricted
```

Lines that begin with `deb` indicate repositories containing binaries. These are used to download precompiled applications and libraries. Lines beginning with `deb-src` are repositories that contain the source code for applications, which we can download and compile or modify ourselves.

The second item on each line is the URL for the repository or the location from which packages will be downloaded.

The third item is the distribution's nickname. On an Ubuntu 16.04 server this will be `xenial`, short for Xenial Xerus. This ensures we're downloading the proper package for our distribution release.

The final items will tell us about support and licensing for the software in the repository. For example, `main` means the packages are stable and supported, and `restricted` means that the licensing may be more restrictive than traditional open source.

Update the System

When our system has repositories installed (i.e., the `sources.list` file is populated), we can use those repositories to update our local list of available packages. The `apt` tool provides a simple way to do this.

To update the system, we use `apt-get`.

```
[linuxacademy@ip] sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates
InRelease [102 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports
InRelease [102 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [102
kB]
Fetched 306 kB in 0s (691 kB/s)
Reading package lists... Done
```

This makes calls to the repositories to see which software packages are available to be installed. The returned lists are stored in our local cache.

Any time you make changes to your repositories, such as adding or removing one from the list, you should run this command. Otherwise, you may end up installing software from an outdated source, leading to version mismatches or other problems.

Third Party Repositories

We can also add third party repositories to be used by `apt`. These can be added to `sources.list` directly, but it's a good idea to separate them from the Debian/Ubuntu repositories by placing them in the `sources.list.d` directory.

To add a third party repository, navigate to the `sources.list.d` directory and create a new file with your text editor of choice.

```
[linuxacademy@ip] cd sources.list.d
[linuxacademy@ip] sudo vim archive.list
```

In that file, add a comment describing the repository and a line similar to the ones in `sources.list`:

```
# adding an archive repository for partner applications
deb http://archive.canonical.com/ubuntu xenial partner
```

Before we can use our newly added repository, we'll need to run `apt-get update` again. This time, there will be a few new lines in the output:

```
Get:5 http://archive.canonical.com/ubuntu xenial InRelease [11.5 kB]
Get:6 http://archive.canonical.com/ubuntu xenial/partner amd64 Packages
[2,668 B]
Get:7 http://archive.canonical.com/ubuntu xenial/partner Translation-en
[1,424 B]
```

Notice the URL matches our addition to `archive.list`. This means the results from that repository have been successfully retrieved and stored in the local cache.

Install Packages

Now that we know how to update our local list of packages, we can start using `apt` to install software. To do so, we'll use `install` rather than `update`, and then provide the name of the package. To see how it works, let's install the `vsftpd` package.

```
[linuxacademy@ip] sudo apt-get install vsftpd
```

We'll see output showing information about dependencies that need to be installed with the package we selected and we'll be prompted before the installation actually runs.

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ssl-cert
Suggested packages:
  openssl-blacklist
The following NEW packages will be installed:
  ssl-cert vsftpd
0 upgraded, 2 newly installed, 0 to remove and 25 not upgraded.
Need to get 132 kB of archives.
After this operation, 398 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

To allow the package and its dependencies to install, enter `y`.

Upgrade Packages

After updating our local cache, we can upgrade any packages that have releases that are newer than the ones currently installed.

```
[linuxacademy@ip] sudo apt-get upgrade
```

This will show a list of packages that can be upgraded, the amount of disk space that will be used when installing them, and a prompt asking you to confirm you want to perform the upgrade. To install the new packages, enter `y` at the prompt.

Upgrade Distribution Version

We can also upgrade the distribution with `apt`. If we're using, for example, Ubuntu 16.04 and want to upgrade to a newer release, we can use `apt-get` to achieve this.

```
[linuxacademy@ip] apt-get dist-upgrade
```

Running the distribution upgrade is optional for the lab; but it's important to be familiar with the command to do so.

Configuration Files

When packaged are installed with `apt`, their configuration files are usually located in the `/etc` directory. Let's take a look at the configuration of the `vsftpd` package we installed earlier.

```
[linuxacademy@ip] cd /etc
[linuxacademy@ip] sudo vim vsftpd.conf
```

In this file, we can make changes to the way the software behaves by adding or modifying lines. Once you've taken a few moments to look through the file, exit back to the command line.

Remove Packages

Earlier, we learned how to use `apt` to install packages, but we can also use it to remove, or uninstall, them. Let's remove the `vsftpd` package from our system.

```
[linuxacademy@ip] sudo apt-get remove vsftpd
```

The output from this command will show some information about the operation, such as the amount of disk space that will be freed, and we'll be prompted to confirm the action. Enter `y` to continue.

Next, let's see the effects this had on the configuration file.

```
[linuxacademy@ip] ls -al /etc/vsftpd.conf
-rw-r--r-- 1 root root 5850 Apr 13  2016 /etc/vsftpd.conf
```

Notice that the file still exists. The `apt-get remove` operation will uninstall the software associated with a package, but not its configuration file. This can be useful if we ever need to reinstall a package; since `apt` doesn't automatically remove the configuration, we won't need to configure it from scratch.

Purge Packages

If you do want to remove the software and its configuration files, we'll need to use a different method. First, let's reinstall `vsftpd` and make sure the configuration file still exists.

```
[linuxacademy@ip] sudo apt-get install vsftpd
[linuxacademy@ip] ls -al /etc/vsftpd.conf
-rw-r--r-- 1 root root 5850 Apr 13  2016 /etc/vsftpd.conf
```

Next, we'll use the `purge` command to remove the package and delete all of its associated files and directories.

```
[linuxacademy@ip] sudo apt-get purge vsftpd
```

Now when we check for the configuration file, we can see that it's been removed.

```
[linuxacademy@ip] ls -al /etc/vsftpd.conf
ls: cannot access '/etc/vsftpd.conf': No such file or directory
```

Search for Packages

We can also search for packages in the local cache. This is useful if we're trying to install a piece of software, but aren't sure of its exact package name. However, be sure to run an update first to ensure the local cache is up to date.

```
[linuxacademy@ip] sudo apt-get update
[linuxacademy@ip] apt-cache search vsftpd
vsftpd - lightweight, efficient FTP server written for security
vsftpd-dbg - lightweight, efficient FTP server written for security
(debug)
ccze - A robust, modular log coloriser
ftpd - File Transfer Protocol (FTP) server
yasat - simple stupid audit tool
```

The output will be a list of packages and their descriptions that contain text related to your search term (the argument provided to `apt-cache search`).

Get Information About a Package

If we know a package's name (or find it with `apt-cache search`), we can view detailed information about the package using the `show` command.

```
[linuxacademy@ip] apt-cache show vsftpd
```

This will output details such as the software's homepage, where to file bugs, the author and maintainer, and much more.

We can also see more technical details with a similar command, `showpkg`.

```
[linuxacademy@ip] apt-cache showpkg vsftpd
```

This shows details such as version information, a list of dependencies, and what services the package will start, among other things.

Review

The Advanced Package Tool is a powerful utility but with just a few commands we can use it to work with system packages quite easily. All of this information is accessible through graphical applications such as Aptitude and Synaptic, but the LPIC exam will test your knowledge of how to use `apt` from the command line.

Congratulations! You've completed the lab on using the `apt` tool!