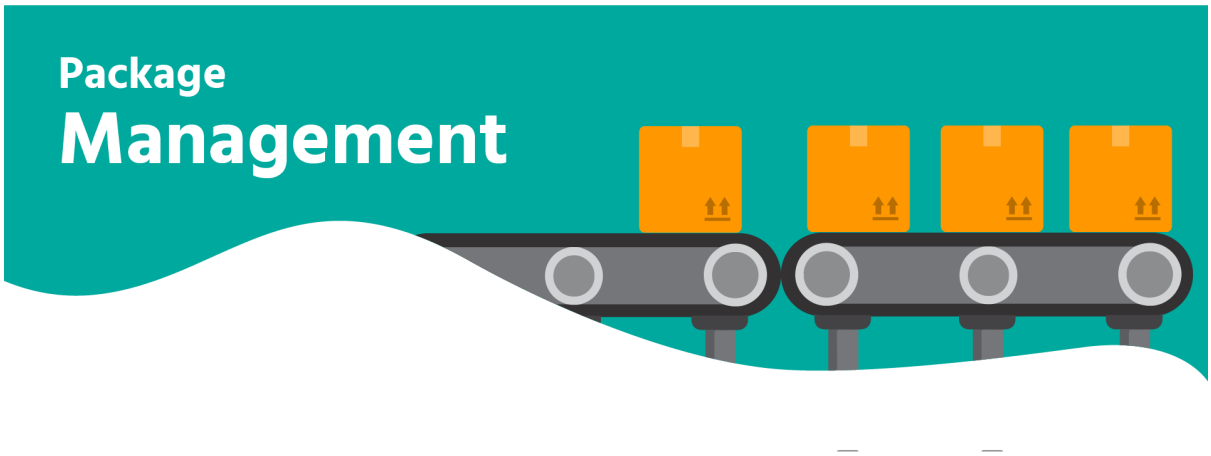
NDG Linux Unhatched - NDG Linux Unhatched

content.netdevgroup.com/contents/unhatched/400



Package Management

Package management is a system by which software can be installed, updated, queried or removed from a filesystem. In Linux, there are many different software package management systems, but the two most popular are those from Debian and Red Hat. The virtual machines for this course use Ubuntu, a derivative of Debian.

At the lowest level of the Debian package management system is the `dpkg` command. This command can be tricky for novice Linux users, so the Advanced Package Tool, `apt-get`, a front-end program to the `dpkg` tool, makes management of packages even easier.

Note

A front-end program is a program that users can see and interact with.

Follow Along

Many of the package management commands require administrative access, so they will be prefaced with the `sudo` command. Use `netlab123` as the password when prompted.

Installing Packages

Package files are commonly installed by downloading them directly from repositories located on Internet servers. The Debian repositories contain more than 65,000 different packages of software. Before installing a package, it is good practice to refresh the list of available packages using the `apt-get update` command.

Please allow a few minutes for the following commands to execute.

```
sudo apt-get update
```

```
sysadmin@localhost:~$ sudo apt-get update
[sudo] password for sysadmin:
Ign file: amd64/ InRelease
Ign file: amd64/ Release.gpg
Ign file: amd64/ Release
Reading package lists... Done
```

To search for keywords within these packages, you can use the `apt-cache search` command.

```
apt-cache search [keyword]
```

The keyword that is used should match part of the name or description of the package that is to be located. Multiple keywords can be used to further clarify the search; for example, the search term `web server` would provide better results than `web` or `server`.

To find packages associated with the `cow` keyword:

```
sysadmin@localhost:~$ apt-cache search cow
cowsay - configurable talking cow
```

Once you've found the package that you want to install, you can install it with the `apt-get install` command:

```
sudo apt-get install [package]
```

```
sysadmin@localhost:~$ sudo apt-get install cowsay
[sudo] password for sysadmin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  filters
The following NEW packages will be installed:
  cowsay
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/18.5 kB of archives.
After this operation, 90.1 kB of additional disk space will be used.

Selecting previously unselected package cowsay.
(Reading database ... 24313 files and directories currently installed.)
Preparing to unpack .../cowsay_3.03+dfsg1-6_all.deb ...
Unpacking cowsay (3.03+dfsg1-6) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up cowsay (3.03+dfsg1-6) ...
```

Consider This

The `cowsay` command is a configurable talking cow! Use a word or phrase as an argument:

```
sysadmin@localhost:~$ cowsay 'NDG Linux Unhatched'
 _____________________
< NDG Linux Unhatched >
 ---------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

We recommend enclosing the argument in single quotes to prevent the shell from interpreting special characters.

Updating Packages

The `apt-get install` command can also update a package, if that package is installed and a newer version is available. If the package is not already on the system, it would be installed; if it is on the system, it would be updated.

Updating all packages of the system should be done in two steps. First, update the cache of all packages available with `apt-get update`. Second, execute the `apt-get upgrade` command and all packages and dependencies will be updated.

```
apt-get update

apt-get upgrade
```

```
sysadmin@localhost:~$ sudo apt-get update
[sudo] password for sysadmin:
Ign file: amd64/ InRelease
Ign file: amd64/ Release.gpg
Ign file: amd64/ Release
Reading package lists... Done
sysadmin@localhost:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Removing Packages

The `apt-get` command is able to either remove or purge a package. The difference between the two is that purging deletes all package files, while removing deletes all but the configuration files for the package.

An administrator can execute the `apt-get remove` command to remove a package or the `apt-get purge` command to purge a package completely from the system.

```
apt-get remove [package]

apt-get purge [package]
```

For example, to purge `cowsay` completely, execute the following command. Enter **Y** when prompted:

```
sysadmin@localhost:~$ sudo apt-get purge cowsay
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  cowsay*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 90.1 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 24377 files and directories currently installed.)
Removing cowsay (3.03+dfsg1-6) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
```