5.1. List Packages

We can ask YUM to list our packages with the following commands:

# List installed and available packages from repositories:

yum list all

# List only installed packages:

yum list installed

# List only available packages:

yum list available

# List installed and available kernel packages:

yum list kernel

With APT, instead, we can list our packages:

# List installed and available packages from repositories:

apt list

# List only installed packages:

apt list --installed

In case we want to see only the available packages, we can resort to some grep since there’s no APT command to do that out of the box:

# List only available packages:

apt list | grep -v installed

For scripting, we could also consider resorting to the nearest solution to apt list available from apt-cache:

apt-cache search .

Beware that the result will differ both in the number of occurrences and in their returned order, though.

5.2. Search Packages

With YUM, we can look for a package containing a specific term in its name or description:

yum search SOMETHING

This time, in APT it’s identical:

apt search SOMETHING

apt-cache search SOMETHING

5.3. Get Package Details

Once we know which package we’re interested in, we might want to check its details.

In YUM:

yum info MY\_PACKAGE

And similarly, in APT:

apt show MY\_PACKAGE

apt-cache show MY\_PACKAGE

6. Package Management

Now that we’ve seen how to search and inspect software packages, let’s see how to manage them:

6.1. Update of Packages Information

First of all, we need to update our package index.

These commands don’t update any installed package, they just download the latest information about the packages that can be installed or upgraded.

In YUM it’s:

yum check-update

In APT, instead, it’s simply:

apt update

apt-get update

We should always run apt update before any other operations.

6.2. Installation of a Package

To install a package in YUM:

yum install MY\_PACKAGE

Again, in Debian systems it doesn’t differ at all:

apt install MY\_PACKAGE

apt-get install MY\_PACKAGE

6.3. Upgrade of a Package

Upgrading a package can be done in different ways.

In YUM, the command yum update internally runs the yum check-update, which means that we don’t need to run the latter unless we want to avoid installing anything after updating the package index. We can upgrade all or some packages as follows:

# Update the package index and every package:

yum update

# Update only the package MY\_PACKAGE:

yum update MY\_PACKAGE

# Apply security-related package updates:

yum update --security

Or:

# Update every package:

apt upgrade

apt-get upgrade

# Update every package along with their dependencies:

apt full-upgrade

apt-get dist-upgrade

# Update only the package MY\_PACKAGE:

apt-get install --only-upgrade MY\_PACKAGE

# Apply security-related package updates:

unattended-upgrade --dry-run -d

It’s important to know that upgrading the packages along with their dependencies potentially implies uninstalling existing software and installing new software as well if this is required by the upgrade process.

Standard upgrade commands, on the other side, will never uninstall anything. However, differently from apt-get upgrade (which also doesn’t install anything), apt upgrade might install new software if needed.

6.4. Removal of a Package

Sometimes we need to remove a software package. Let’s explore the different ways of doing this then, from shallow and soft to deep and final.

To get rid of an installed package and possibly its dependencies in YUM we can do one of two equivalent commands:

yum erase MY\_PACKAGE

yum remove MY\_PACKAGE

In RHEL7 and higher, it’s possible to erase also additional unneeded packages with autoremove:

yum autoremove MY\_PACKAGE

The Debian ways to delete a package instead are:

apt remove MY\_PACKAGE

apt-get remove MY\_PACKAGE

However, if we want to remove the package’s configuration too, completely purging the system from it, then we can exploit purge:

apt purge MY\_PACKAGE

apt-get purge MY\_PACKAGE

6.5. Clean Up

Sometimes, our system will be polluted by orphaned packages, which are not needed anymore but are still occupying space.

We can remove these unwanted packages in YUM through autoremove, without any package name:

yum autoremove

This also works in the same way on Debian distributions:

apt autoremove

apt-get autoremove

7. Repository Management

Both packaging systems start with a set of official repositories to query for fetching packages.

However, the community is thriving, and often the package we need is missing in the official repositories, or is there but in a version too old to fit our needs.

In these cases, we might want to add unofficial repositories to the package manager list (always paying attention to the fact that it might represent a security issue).

7.1. Addition of a Repository

Adding a repository in YUM is a manual operation, which consists in creating a file with the .repo extension under the folder /etc/yum.repos.d.

The file must contain all the information about the custom repository that we are connecting to.

Let’s try adding the AdoptOpenJDK repository:

# /etc/yum.repos.d/adoptopenjdk.repo

[AdoptOpenJDK]

name=AdoptOpenJDK

baseurl=http://adoptopenjdk.jfrog.io/adoptopenjdk/rpm/centos/7/$(uname -m)

enabled=1

gpgcheck=1

gpgkey=https://adoptopenjdk.jfrog.io/adoptopenjdk/api/gpg/key/public

In APT, though, things are quite different. The GPG key of the repository must be downloaded and added to the APT keyring with apt-key add:

wget -qO - https://adoptopenjdk.jfrog.io/adoptopenjdk/api/gpg/key/public | sudo apt-key add -

Then, at this point, the repository can be added through add-apt-repository –yes followed by the URL:

add-apt-repository --yes https://adoptopenjdk.jfrog.io/adoptopenjdk/deb/

Contrary to YUM, all the repositories are saved in a single file, /etc/apt/sources.list.

7.2. Removal of a Repository

Removing a repository in YUM is performed differently depending on how it’s been installed.

We can run the following command and analyze its output:

rpm -qa | grep -i CUSTOM\_REPOSITORY

If the repository’s RPM package is found, it means it’s been installed through RPM, and we can remove it using -e:

rpm -e CUSTOM\_REPOSITORY\_RPM\_PACKAGE

Otherwise, we can simply delete the repository file:

rm /etc/yum.repos.d/CUSTOM\_REPOSITORY.repo

We can also disable it without deleting it, by simply turning enabled=1 to enabled=0 in the repository file.

In APT, on the other hand, we can simply do:

add-apt-repository --remove ppa:CUSTOM\_REPOSITORY/ppa

Alternatively, we can comment out the rows relative to the repository in the /etc/apt/sources.list file.

8. Other Operations

These two do a lot of other operations. Let’s cover some of them.

8.1. Reinstalling a Package

In case of a corrupted package, like some files are missing, we can reinstall it with:

yum reinstall MY\_PACKAGE

apt reinstall MY\_PACKAGE

apt-get reinstall MY\_PACKAGE

8.2. Installing a Specific Version of a Package

Sometimes, instead, we might need to install a specific version of a package.

We can check the available versions of a package with:

yum list MY\_PACKAGE --showduplicates

apt-cache showpkg MY\_PACKAGE

Then we can target the wanted version for a new installation:

yum install MY\_PACKAGE-WANTED\_VERSION

apt install MY\_PACKAGE=WANTED\_VERSION

apt-get install MY\_PACKAGE=WANTED\_VERSION

YUM also allows us to downgrade from an existing one to one of the previously installed versions:

yum downgrade MY\_PACKAGE-WANTED\_VERSION

8.3. Installing a Group of Packages

Some packages are combined in a group for a common purpose and can be installed all at once.

Let’s target another real-world example and think of installing the graphics environment in a Linux server.

Installing the X Window System and GNOME groups will save us the hassle of installing hundreds of packages by hand:

yum groupinstall 'X Window System' 'GNOME'

The Debian package manager, however, handles them as simple packages, so the classic apt install command is enough:

apt-get install xorg