**How to install packages on RHEL 8 / CentOS 8 Linux System**

Egidio Docile

Redhat / CentOS

03 April 2019

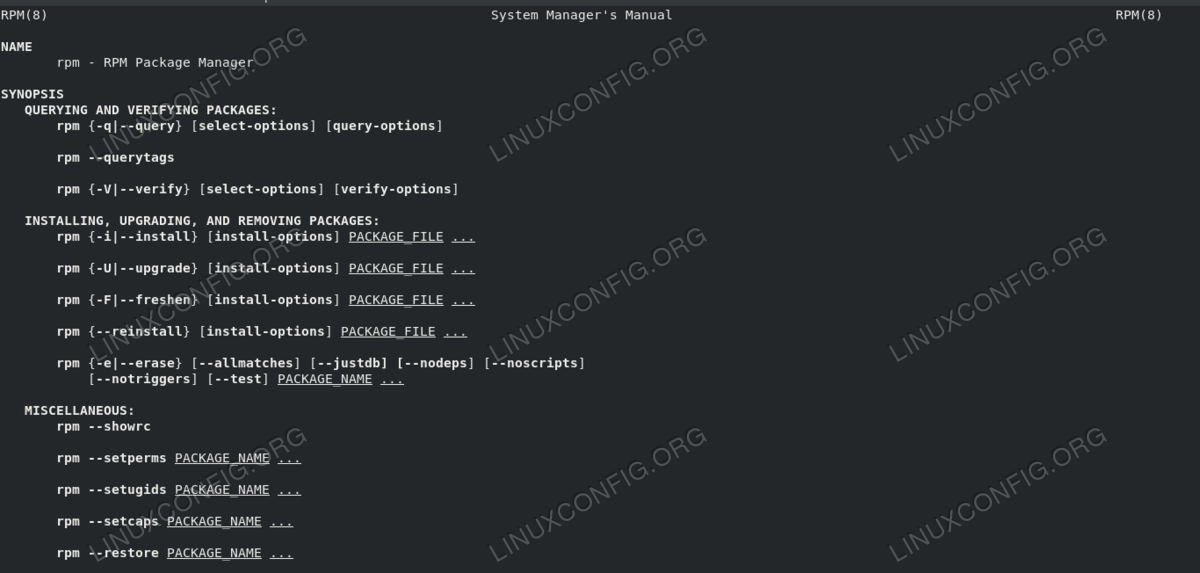
**Contents**

* + [1. Software Requirements and Conventions Used](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h1-software-requirements-and-conventions-used)
  + [2. What is a package?](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h2-what-is-a-package)
  + [3. RPM: The Red Hat Package Manager and yum/dnf](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h3-rpm-the-red-hat-package-manager-and-yum-dnf)
  + [4. Installing a package with rpm](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h4-installing-a-package-with-rpm)
    - * [4.1.1. Installing a package from a remote location](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h4-1-1-installing-a-package-from-a-remote-location)
  + [5. Installing a package with dnf](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h5-installing-a-package-with-dnf)
  + [6. Installing a package graphically](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h6-installing-a-package-graphically)
  + [7. Installing software with flatpacks](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h7-installing-software-with-flatpacks)
  + [8. Conclusions](https://linuxconfig.org/how-to-install-packages-on-redhat-8#h8-conclusions)

All modern linux distributions organize software in packages which contain applications binaries, files, metadata and information about the package dependencies, possible conflicts with other packages etc. The core Rhel package manager is called rpm itself, and it is the tool used also by dnf, the higher level package manager, which is capable to manage dependencies. A relatively recent technology , flatpak, let us install also sandboxed applications together with their runtimes.

**In this tutorial you will learn:**

* The three main ways to install a package using rpm
* How to install packages using dnf
* How to install applications graphically from the gnome-software utility
* How to install sandboxed applications with flatpak

[](https://linuxconfig.org/images/rpm_manpage.png)

The rpm package manager manual

**Software Requirements and Conventions Used**

| *Software Requirements and Linux Command Line Conventions* | |
| --- | --- |
| **Category** | **Requirements, Conventions or Software Version Used** |
| **System** | RHEL 8 / CentOS 8 |
| **Software** | The utilities used in this tutorial are installed by default, there is no need for additional software. |
| **Other** | Permission to run command with root privileges. |
| **Conventions** | **#** - requires given [linux commands](https://linuxconfig.org/linux-commands) to be executed with root privileges either directly as a root user or by use of sudo command **$** - requires given [linux commands](https://linuxconfig.org/linux-commands) to be executed as a regular non-privileged user |

**What is a package?**

In [RHEL 8](https://linuxconfig.org/how-to-install-rhel-8) / CentOS 8, like in all modern linux distributions, software is organized in repositories, in the form of packages. But what exactly is a package? A package is basically a compressed archive which contains the files associated with a specific software, like compiled binaries and libraries, plus metadata and instructions about where and how those files should be installed on the system, the external dependencies the software needs to run correctly and more. RHEL and CentOS uses the rpm package format which was invented by Red Hat itself: in this tutorial we will see how to install and manage rpm packages on Red Hat Enterprise Linux 8.

***SUBSCRIBE TO NEWSLETTER*** *Subscribe to Linux Career*[*NEWSLETTER*](https://bit.ly/2X5D30q)*and receive latest Linux news, jobs, career advice and tutorials.*

**RPM: The Red Hat Package Manager and yum/dnf**

To ease the management of rpm packages, the rpm package manager was created. Although very powerful, it is focused on single packages, so it is not able to resolve dependencies. To manage the relationships between packages, and avoid the infamous dependency hell, first yum package manager (Yellow Updater Modified) and then its evolution, dnf (Dandified YUM), were adopted. Dnf was firstly introduced in Fedora 18, and does replace yum in the last version of Red Hat Enterprise Linux.

**Installing a package with rpm**

One of the most basic operations we can perform with rpm is the installation of a package. As said before, this will work only if the package has no external dependencies, since rpm is not able to manage it. There are basically three ways we can install a package with rpm: they correspond respectively to the -i, -U and -F options.

To install a new package we can run rpm with the -i option (short for --install). The syntax is very simple:

$ sudo rpm -i *packagename*.rpm

When we run the command above, the specified package is installed only if not already installed on the system. If we try to install an already installed package (gedit in this case), we receive an error:

$ sudo rpm -i gedit-3.28.1-3.el8.x86\_64.rpm

package gedit-2:3.28.1-3.el8.x86\_64 is already installed

Notice that here for same "package" we mean same exact version. This method of installation is the recommended one to use when performing the installation of a new kernel: in such situations, the new kernel will coexist with the older one(s), so in case something doesn't work as expected, it will be possible to use them as a fallback.

Things go differently when we use the -U option, which is the short for --upgrade. In this case if a package doesn't already exist in the system, it is installed; if a previous version of the package exists, the package is upgraded. This means that after the newer version of the package is installed, the older ones are removed. As we said above, upgrading a fundamental piece of software like the kernel in place is highly discouraged: it's better to be absolutely sure that the new kernel works as expected before removing older versions.

The -F (or --freshen) option does something different. When it is used, only already existing packages are updated. If a previous version of package we are trying to install doesn't already exist in our system, it is not installed.

**Installing a package from a remote location**

With rpm is also possible to install a package which is not on our local filesystem. Say for example we want to install a package accessible via the ftp protocol; the command to run would be the following:

$ sudo rpm -i ftp://ftp.someaddress.com/package.rpm

It is also possible to provide an username and a password if authentication is required, including such information in the address:

$ sudo rpm -i ftp://username:password@ftp.someaddress.com/package.rpm

**Installing a package with dnf**

Although dnf operates on top of rpm, it provides very useful features like the management of software dependencies, and the ability to install software from the distribution repositories. The metadata stored in a rpm package, among other things, specifies what other packages should be installed for the software to work, and, eventually, what packages could conflict with it. Dnf, or takes care of performing all the needed operations, after receiving user confirmation. Say for example we want to install the vim package using dnf; we would run:

$ sudo dnf install vim

After running the command above, the operations to be performed and the list of dependencies to be installed are displayed on the screen:

Dependencies resolved.

================================================================================

Package Arch Version Repository Size

================================================================================

Installing:

vim-enhanced

x86\_64 2:8.0.1763-7.el8 rhel-8-for-x86\_64-appstream-beta-rpms 1.4 M

Installing dependencies:

gpm-libs x86\_64 1.20.7-15.el8 rhel-8-for-x86\_64-appstream-beta-rpms 39 k

vim-filesystem

noarch 2:8.0.1763-7.el8 rhel-8-for-x86\_64-appstream-beta-rpms 47 k

vim-common x86\_64 2:8.0.1763-7.el8 rhel-8-for-x86\_64-appstream-beta-rpms 6.3 M

Transaction Summary

================================================================================

Install 4 Packages

Total download size: 7.8 M

Installed size: 30 M

Is this ok [y/N]:

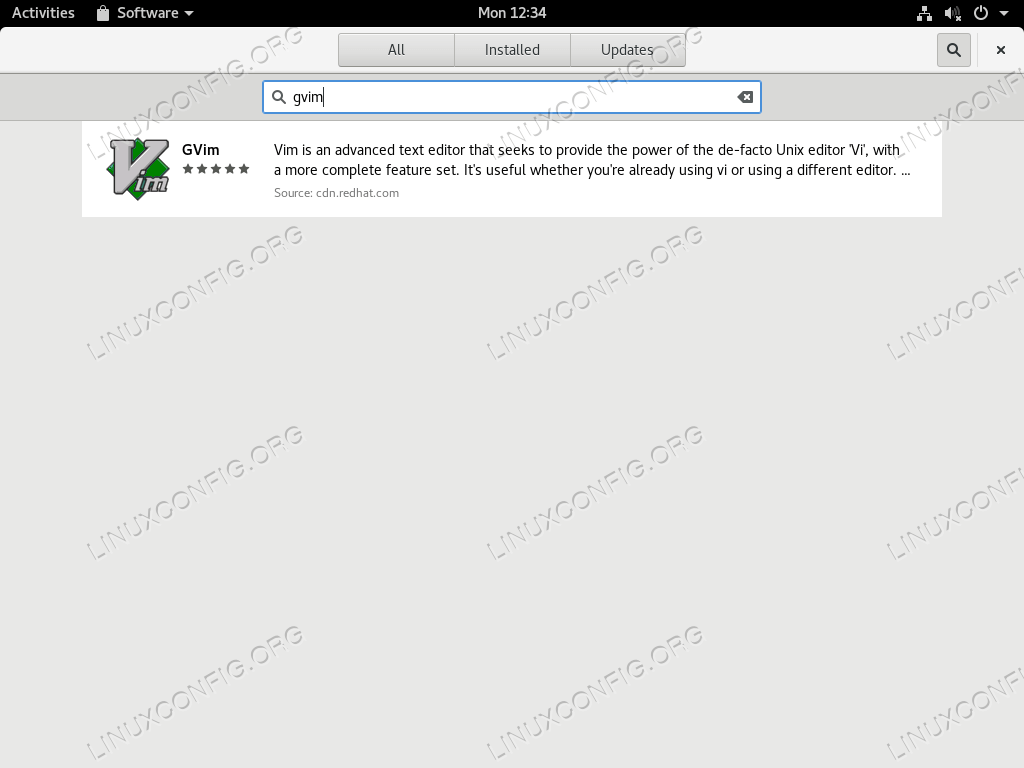
If running dnf from a script which is meant to run unattended, the -y option (short form of --assumeyes) can be used: that way all the answers will receive a 'yes' response automatically, and user interaction will be required.

Like rpm, dnf let us install a package located on a remote url; all we have to do is to specify the latter:

$ sudo dnf install http://path/to/package.rpm

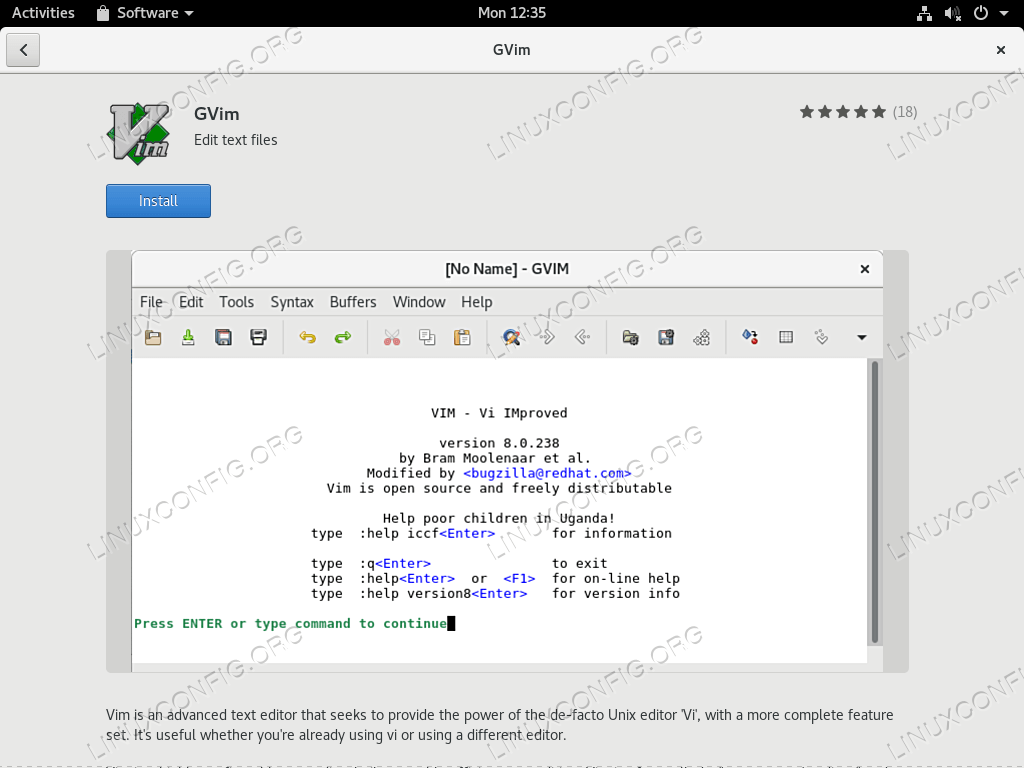
**Installing a package graphically**

When running the GNOME desktop environment, is also possible to install a package graphically, using the gnome-software application. If the package exists in the distribution repositories we can simply perform a search for the application we want to install using the search field:

[](https://linuxconfig.org/images/rhel8-gnome-software-search.png)

Gnome software search interface

After that, we can click on the "install" button to actually install the application:

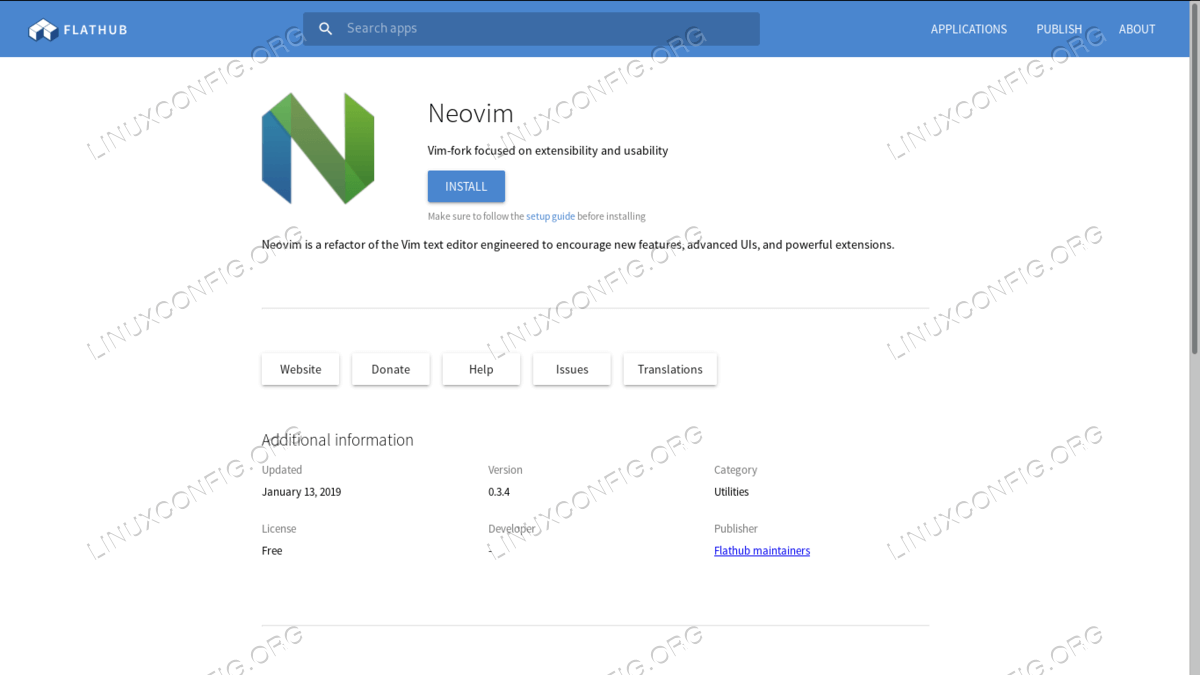
[](https://linuxconfig.org/images/rhel8-gnome-software-install.png)

Gnome software installation interface

When installing a package with this method a different procedure is used. The gnome-software applications is a frontend for packagekit which is itself a high-level frontend which works as an abstraction over the native package manager of a distribution: its main goal is to uniform installations on different distributions. It's also important to notice that packages installed this way will not appear in the dnf history, since packagekit doesn't use the latter under the hood.

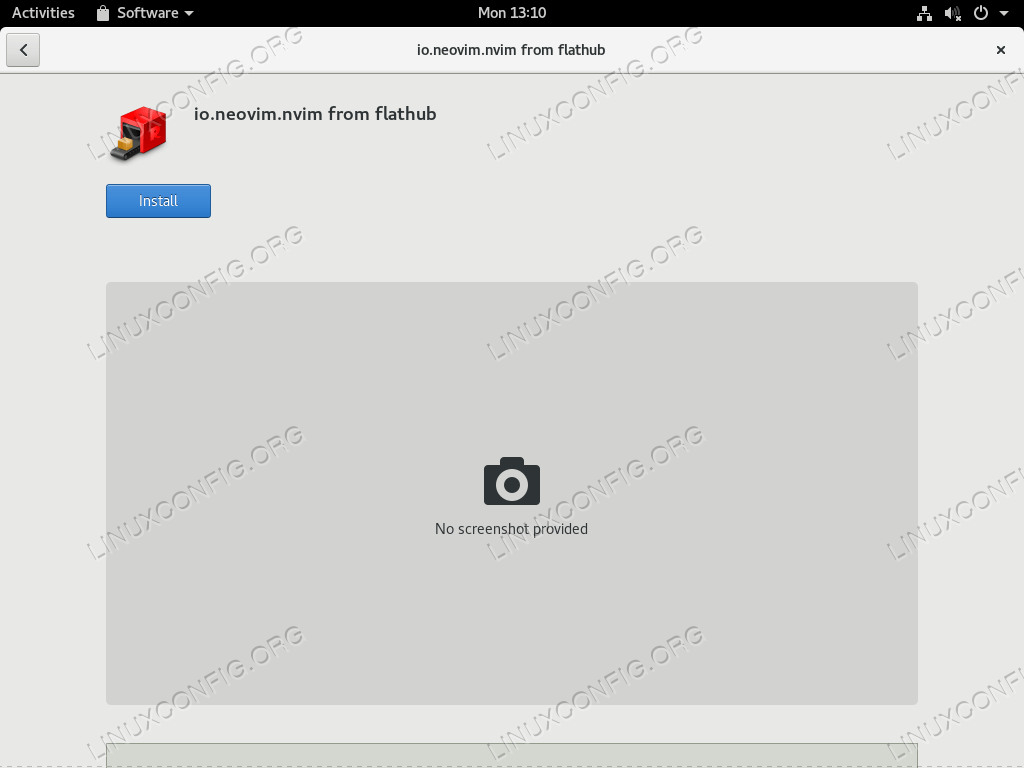
**Installing software with flatpacks**

On [RHEL 8](https://linuxconfig.org/how-to-install-rhel-8) / CentOS 8, another possible way to install software, both from the gnome-software application, or from command line, is to use flatpacks. Flatpacks provide sandboxed environments inside which a software is provided together with all its dependencies: this makes possible to run a specific version of a program on all the distributions which support this technology, independently of the native libraries. The main repository of flatpacks is [flathub](https://flathub.org/home). Installing a flatpack from this site with gnome-software is really easy. All we have to do is to search for the application we want to install and click on the install button.

[](https://linuxconfig.org/images/flathub-interface.png)

Flathub is the main repository used for flatpaks

Gnome software should be automatically launched: if not, just open the downloaded file with it. After we click on the install button, and provide our administrative password, if required, the package will be installed.

[](https://linuxconfig.org/images/gnome-software-flatpak-install.png)

Gnome software flatpak installation

Once we install a flatpak this way, the flathub repository is automatically added as a flatpak source, therefore it becomes possible to search for other applications contained in it, directly from the gnome-software center interface.

It is possible to perform the same operations we saw above from the command line using the flatpak command. When using the flatpak utility, it's possible to decide if the installation should be globally or just for a specif user. In the latter case administrative privileges are not required, and the --user option must be provided to the command. To add the flathub repository globally we simply run:

$ sudo flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo

Now we can install our package. We must provide the full package path and also the name of the remote repository we want to install from. For example to install neovim we would run:

$ sudo flatpak install flathub io.neovim.nvim

Required runtime for io.neovim.nvim/x86\_64/stable (runtime/org.freedesktop.Sdk/x86\_64/18.08) found in remote flathub

Do you want to install it? [y/n]:

If we answer affirmatively the package and its runtime will be downloaded and installed in our system. If you want to know more about the flatpak technology, you can checkout our more detailed [article on the subject](https://linuxconfig.org/how-to-use-flatpak-to-manage-sandboxed-applications-on-fedora-workstation-28)

**Conclusions**

In this tutorial we saw various ways to install software on RHEL 8 / CentOS 8 Linux Systems. Software is organized in packages, which at a low level are managed by the rpm package manager which cannot resolve dependencies automatically. To provide this feature and more, the dnf program, which works on top of rpm, is used. Finally we learned to install sandboxed software using a relatively recent technology, flatpak.