

# RHSA (Admin 1) Red Hat Enterprise Linux 8 Essentials

Prepared by Eng. Taghreed Elsayed

## Lecture 9

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي  
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

Installing and updating software packages



جامعة الجميع الذكية  
EVERYONE'S SMART UNIVERSITY

# Installing and updating software packages

## What is a package?

- ❑ In [RHEL 8](#) / CentOS 8, like in all modern linux distributions, software is organized in repositories, in the form of packages.
- ❑ **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.



# Installing and updating software packages

## RPM: The Red Hat Package Manager and yum

- ☐ 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,



# Installing and updating software packages

## 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**).



# Installing and updating software packages

- The syntax is very simple:

```
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:

```
rpm -i gedit-3.28.1-3.el8.x86_64.rpm  
package gedit-2:3.28.1-3.el8.x86_64 is already installed
```



## Installing and updating software packages

- ❑ 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.



## Installing and updating software packages

- ❑ 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 and updating software packages

### 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:

```
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:

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



## Installing and updating software packages

- In RHEL 8, software installation is enabled by the new version of the **YUM** tool (**YUM v4**), which is based on the **DNF** technology.

### Note

1. Upstream documentation identifies the technology as **DNF** and the tool is referred to as **DNF** in the upstream. As a result, some output returned by the new **YUM** tool in RHEL 8 mentions **DNF**.
- ❑ Although **YUM v4** used in RHEL 8 is based on **DNF**, it is compatible with **YUM v3** used in RHEL 7. For software installation, the `yum` command and most of its options work the same way in RHEL 8 as they did in RHEL 7.



## Installing and updating software packages

- ❑ Selected **yum** plug-ins and utilities have been ported to the new DNF back end, and can be installed under the same names as in RHEL 7.
- ❑ Packages also provide compatibility symlinks, so the binaries, configuration files, and directories can be found in usual locations.

→→ **Note that the legacy Python API** provided by **YUM v3** is no longer available. You can migrate your plug-ins and scripts to the new API provided by **YUM v4** (DNF Python API), which is stable and fully supported.



# Installing and updating software packages

## Application streams

- ❑ RHEL 8 introduces the concept of Application Streams. Multiple versions of user space components are now delivered and updated more frequently than the core operating system packages.
- ❑ This provides greater flexibility to customize Red Hat Enterprise Linux without impacting the underlying stability of the platform or specific deployments.
- ❑ Components made available as Application Streams can be packaged as modules or RPM packages, and are delivered through the **AppStream repository in RHEL 8**.
- ❑ Each Application Stream has a given life cycle, either the same as RHEL 8 or shorter, more suitable to the particular application. Application Streams with a shorter life cycle are listed in the [Red Hat Enterprise Linux 8 Application Streams Life Cycle](#) page.



## Installing and updating software packages

- ❑ Modules are collections of packages representing a logical unit: an application, a language stack, a database, or a set of tools. These packages are built, tested, and released together.
- ❑ Module streams represent versions of the Application Stream components. For example, two streams (versions) of the **PostgreSQL database server** are available in the postgresql module: PostgreSQL 10 (the default stream) and PostgreSQL 9.6.
- ❑ Only one module stream can be installed on the system. Different versions can be used in separate containers.
- ❑ Detailed module commands are described in the [Installing, managing, and removing user-space components](#) document. For a list of modules available in AppStream, see the [Package manifest](#).



## Searching for software packages

❑ **yum** allows you to perform a complete set of operations with software packages.

The following section describes how to use **yum** to:

Search for packages.

1. List packages.
2. List repositories.
3. Display information about the packages.
4. List package groups.
5. Specify global expressions in yum input.



## 1. Searching packages with YUM

- ❑ Use the following procedure to find a package providing a particular application or other content.

### ➤ Procedure

- To search 

```
# yum search term
```

- ❑ Replace *term* with a term related to the package.
  - ❑ **Note that `yum search command`** returns term matches within the name and summary of the packages. This makes the search faster and enables you to search for packages you do not know the name of, but for which you know a related term.



# Installing and updating software packages

- ❑ To include term matches within package descriptions, use:

```
# yum search --all term
```

- ❑ Replace *term* with a term you want to search for in a package name, summary, or description.

**Note that yum search --all** enables a more exhaustive but slower search.

## 2.Listing packages with YUM

- ❑ Use the following procedure to list installed and available packages.

### Procedure

- ❑ To list information on all installed and available packages, use:

```
# yum list --all
```

- To list all packages installed on your system, use:

```
# yum list --installed
```

- To list all packages in all enabled repositories that are available to install, use:

```
# yum list --available
```

## 3. Listing repositories with YUM

❑ Use the following procedure to list enabled and disabled repositories.

### Procedure

- To list all enabled repositories on your system, use:

**# yum repolist**

- To list all disabled repositories on your system, use:

**# yum repolist –disabled**

- To list both enabled and disabled repositories, use:

**# yum repolist –all**



# Installing and updating software packages

## 4. Displaying package information with YUM

- ❑ You can display various types of information about a package using YUM, for example version, release, size, loaded plugins, and more.'

### ❑ Procedure

- To display information about one or more packages, use:

```
# yum info package-name
```

Replace *package-name* with the name of the package.



# Installing and updating software packages

## 5. Listing package groups with YUM

- ❑ Use yum to view installed package groups and filter the listing results.

### Procedure

- To view the number of installed and available groups, use:

```
# yum group summary
```

- To list all installed and available groups, use:

```
# yum group list
```

❖ Note that you can filter the results by appending command line options for the yum group list command (--hidden, --available). For more available options see the man pages.



# Installing and updating software packages

- To list mandatory and optional packages contained in a particular group, use:

```
# yum group info group-name
```

Replace *group-name* with the name of the group.

## ❖ Installing a package group with YUM

- The following procedure describes how to install a package group by a group name or by a groupID using yum.

### Procedure

To install a package group by a group name, use:



# Installing and updating software packages

- To install a package group by a group name, use:

```
# yum group install group-name
```

Or

```
# yum install @group-name
```

Replace *group-name* with the full name of the group or environmental group.

## Specifying a package name in YUM input

- ❑ To optimize the installation and removal process, you can append `-n`, `-na`, or `-nevra` suffixes to `yum install` and `yum remove` commands to explicitly define how to parse an argument:

## Installing and updating software packages

- To install a package using its exact name, use:

**# yum install-n *name***

Replace *name* with the exact name of the package.

- To install a package using its exact name and architecture, use:

**# yum install-na *name.architecture***

Replace *name* and *architecture* with the exact name and architecture of the package.

- To install a package using its exact name, epoch, version, release, and architecture, use:

**# yum install-nevra *name-epoch:version-release.architecture***

Replace *name*, *epoch*, *version*, *release*, and *architecture* with the exact name, epoch, version, release, and architecture of the package.



# Installing and updating software packages

## Updating software packages

- ❑ **yum** allows you to check if your system has any pending updates. You can list packages that need updating and choose to update a single package, multiple packages, or all packages at once.
- ❑ If any of the packages you choose to update have dependencies, they are updated as well.
- **The following section describes how to use yum to:**
  1. Check for updates.
  2. Update a single package.
  3. Update a package group.
  4. Update all packages and their dependencies.

# Installing and updating software packages

## 1. Checking for updates with YUM

- ❑ The following procedure describes how to check the available updates for packages installed on your system using yum.

### ➤ Procedure

- To see which packages installed on your system have available updates, use:

```
# yum check-update
```

- The output returns the list of packages and their dependencies that have an update available.



## 2. Updating a single package with YUM

- ❑ Use the following procedure to update a single package and its dependencies using yum.

- To update a package, use:

**# yum update *package-name***

- ❑ Replace *package-name* with the name of the package.

→ **Important**

- ❑ When applying updates to kernel, **yum** always **installs** a new kernel regardless of whether you are using the yum update or yum install command.



# Installing and updating software packages

## 3. Updating a package group with YUM

- ❑ Use the following procedure to update a group of packages and their dependencies using yum.

### ❖ Procedure

- **To update a package group, use:**

**# yum group update *group-name***

Replace *group-name* with the name of the package group.



## 4. Updating all packages and their dependencies with YUM

- ☐ Use the following procedure to update all packages and their dependencies using yum.

### ❖ Procedure

- To update all packages and their dependencies, use:  
**# yum update**

## Installing Packages on RHEL 8 Locally Using ISO

### ❑ Benefits of Using RHEL 8 ISO as a Local Repository

- ❖ Allows you to apply offline patches.
- ❖ You can update your server in critical cases where you don't have an internet connection.
- ❖ Using local repo, you can create an [ultra-secure RHEL 8 environment](#) that will never be connected to the internet and still be updated with the most frequent packages.
- ❖ You can also upgrade the server from RHEL 8.x to RHEL 8.y.



## Installing and updating software packages

- ❑ Often, when we want to have a local repository for our **RHEL 8** system to install packages without internet access for extra safety and using **RHEL 8** ISO is the easiest way to do that.
- ❑ In this guide, we will be showing you how you can configure and use locally downloaded **RHEL 8 ISO** images as a repository for installing packages in **RHEL 8** Linux.  
[ You might also like: [How to Create Local RHEL 9 Repository](#) ]
- ❑ But before that, let's have a look at what are the advantages of using **RHEL 8** ISO as a local repository.



# Installing and updating software packages

## Installing Packages on RHEL 8 Locally Using ISO

### Step 1: Download RHEL 8 ISO Image

- ☐ You can easily [download RHEL 8 ISO file](#) from Red Hat's official download page. While downloading ISO, make sure you avoid boot ISOs as they don't include packages that are less than 1GB in size.

### Step 2: Mount RHEL 8 DVD ISO

- ☐ Once we are done downloading the ISO file, we have to create a mounting point for mounting the recently downloaded ISO file on our system.
- ☐ We will be creating a mounting point under the `/mnt` directory by using the given command:



# Installing and updating software packages

```
mkdir -p /mnt/disc  
mount -o loop rhel-8.6-x86_64-dvd.iso /mnt/disc
```

```
[sagar@tecmint Downloads]$ sudo mount -o loop rhel-8.6-x86_64-dvd.iso /mnt/disc  
mount: /mnt/disc: WARNING: device write-protected, mounted read-only.  
[sagar@tecmint Downloads]$
```

Mount RHEL 8 DVD

- ❑ Make sure you change the name of the ISO file before mounting or it will bring an error! You might be wondering about the warning it gave us while we mounted our ISO but don't worry, we will change permissions in the latter part of this guide.

## Installing and updating software packages

```
mkdir -p /mnt/disc  
mount /dev/sr0 /mnt/disc
```

While mounting, make sure you change **sr0** with your drive name.

### **Step 3: Creating Local RHEL 8 Repository**

- ❑ After mounting **RHEL 8** ISO file at **/mnt**, we need to get a copy of the **media.repo** file and paste it to our system directory located at **/etc/yum.repos.d/** with the name of **rhel8.repo**.



## Installing and updating software packages

```
cp /mnt/disc/media.repo /etc/yum.repos.d/rhel8.repo
```

- ❑ As we mentioned earlier, our drive is write-protected. But for our purpose, we have to change permissions of **rhel8.repo** file to **0644** which will allow us to read and make changes accordingly.

```
chmod 644 /etc/yum.repos.d/rhel8.repo
```

## Installing and updating software packages

- ❑ Once we are done with changing permissions, we have to make some changes to make our local repository work. First, let's open **rhel8.repo** file by utilizing the given command:

```
vi /etc/yum.repos.d/rhel8.repo
```

- ❖ Clear the default configuration and paste new instructions in the config file as given below:



# Installing and updating software packages

```
[dvd-BaseOS]
name=DVD for RHEL - BaseOS
baseurl=file:///mnt/disc/BaseOS
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[dvd-AppStream]
name=DVD for RHEL - AppStream
baseurl=file:///mnt/disc/AppStream
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

# Installing and updating software packages

The final result config file will look like this:

```
GNU nano 2.9.8 /etc/yum.repos.d/rhel8.repo

[dvd-BaseOS]
name=DVD for RHEL - BaseOS
baseurl=file:///mnt/disc/BaseOS
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[dvd-AppStream]
name=DVD for RHEL - AppStream
baseurl=file:///mnt/disc/AppStream
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[ Read 13 lines ]
^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos    M-U Undo
^X Exit          ^R Read File    ^\ Replace     ^U Uncut Text  ^T To Spell   ^_ Go To Line  M-E Redo
```

# Installing and updating software packages

- ❑ After configuring the file, we need to clear the **YUM** cache by using the given [yum](#) or [dnf](#) command:

```
yum clean all
```

or

```
dnf clean all
```

- ❑ Now, let's list the enabled repositories on our system by the given command:



# Installing and updating software packages

```
yum repolist enabled
```

```
[sagar@tecmint ~]$ sudo dnf repolist enabled
Updating Subscription Management repositories.
repo id                repo name
dvd-AppStream           DVD for RHEL - AppStream
dvd-BaseOS              DVD for RHEL - BaseOS
rhel-8-for-x86_64-appstream-rpms Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
rhel-8-for-x86_64-baseos-rpms  Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
```

List RHEL 8 Local Repository

So if you went through the process as we mentioned, you will get the output listing two additional repositories named “**dvd-AppStream**” and “**dvd-BaseOS**” which means we have successfully converted our ISO into a local repository.



# Installing and updating software packages

Now, let's update the repository index by using the given command:

```
yum update
```

```
dnf update
```





جامعة الجميع الذكية  
EVERYONE'S SMART UNIVERSITY

