

## 13.3 Paquetes de software RPM y yum

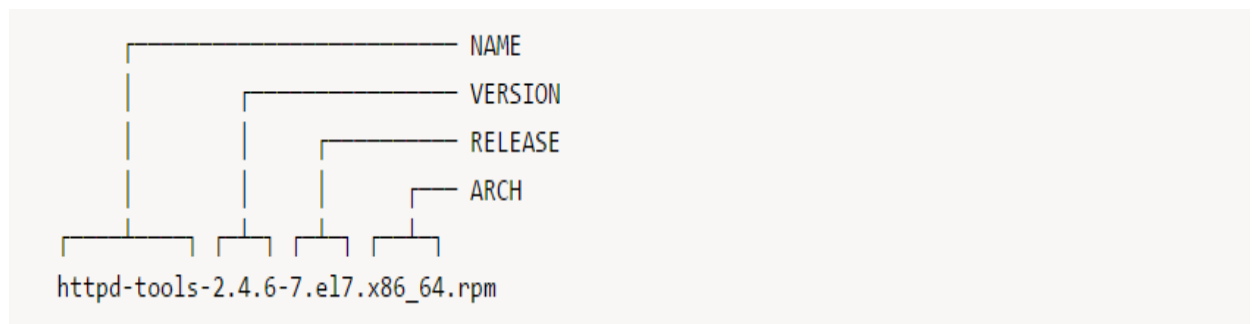
### Objetivos

Tras finalizar esta sección, los estudiantes deberían poder explicar qué es un paquete RPM y cómo se utilizan los paquetes de este tipo para administrar software en un sistema con Red Hat Enterprise Linux.

### Paquetes de software y RPM

Hace varios años, Red Hat desarrolló RPM Package Manager, que ofrece una manera estándar de colocar software en paquetes para su distribución. Administrar software en forma de *paquetes RPM* es mucho más sencillo que trabajar con un software que simplemente se extrajo de un fichero en un sistema de archivos. Esta solución permite que los administradores controlen qué archivos se instalaron con el paquete de software y cuáles deben quitarse si el software se desinstala, y que verifiquen que los paquetes compatibles estén presentes cuando se instale. La información sobre los paquetes instalados se almacena en una base de datos de RPM local en cada sistema. Todo el software proporcionado por Red Hat para Red Hat Enterprise Linux se ofrece como un paquete RPM.

Los archivos de paquetes RPM reciben su nombre de acuerdo con una combinación del paquete `name-version-release.architecture`:



- NAME es una o más palabras que describen los contenidos (`httpd-tools`).
- VERSION es el número de versión del software original (`2.4.6`).
- RELEASE es el número de lanzamiento del paquete que se basa en la versión y que es definido por el empaquetador, que es posible que no sea el desarrollador del software original (`7.el7`).

- ARCH es la arquitectura de procesador en la que se compiló el paquete para su ejecución. "noarch" indica que los contenidos de este paquete no es específico de la arquitectura (x86\_64).

Cuando se instalan paquetes de repositorios, solo se necesita el nombre de paquete. Se instalará el paquete con la versión superior. Si hay múltiples archivos con la misma versión, se instalará el paquete con el número de lanzamiento superior.

Cada paquete RPM es un fichero especial con tres componentes:

- Los archivos instalados con el paquete.
- Información sobre el paquete (metadatos), como el nombre, la versión, el lanzamiento y la arquitectura; un resumen y una descripción del paquete; determinación de si se necesita instalar otros paquetes; licencias; un registro de cambio del paquete; y otros detalles.
- Scripts que pueden ejecutarse cuando el paquete se instala, actualiza o quita, o que se activan cuando otros paquetes se instalan, actualizan o quitan.

Los paquetes RPM pueden llevar la firma digital de la organización que los colocó en paquete. Todos los paquetes de un origen particular llevan normalmente la firma de la misma clave privada GPG. Si el paquete se modifica o daña, la firma dejará de tener validez. De esta manera, el sistema podrá verificar la integridad de los paquetes antes de instalarlos. Todos los paquetes RPM lanzados por Red Hat llevan firma digital.

### Actualizaciones y parches

Cuando Red Hat revisa el código fuente ascendente de un paquete de software, se genera un paquete RPM completo. Si un paquete se añade a un sistema, se necesita solo la última versión del paquete y no cada versión del paquete desde el primer lanzamiento. En el caso de sistemas que necesitan actualizaciones, se quita la versión anterior del paquete y se instala la versión nueva. Los archivos de configuración generalmente se conservan durante una actualización, pero el comportamiento exacto de un paquete en particular se define cuando se crea la nueva versión del paquete.

En la mayoría de los casos, solo una versión o un lanzamiento de un paquete puede instalarse a la vez. Generalmente, el proceso de instalación de RPM no permitirá que los archivos se sobrescriban. Si un paquete se crea de modo que no haya nombres de archivos en conflicto, múltiples versiones pueden instalarse. Este es el caso del paquete `kernel`. Como un kernel nuevo solo puede evaluarse mediante el inicio en ese núcleo, el paquete está específicamente diseñado para que puedan instalarse múltiples versiones simultáneamente. Si el kernel nuevo no arranca, el kernel anterior sigue estando disponible.

## Administrador de paquetes yum

Una vez que se instala un sistema, los paquetes y las actualizaciones de software adicionales normalmente se instalan desde un *repositorio de paquetes* de red, la mayoría de las veces a través del servicio de administración de suscripciones de Red Hat que se analizó en la sección anterior. El comando **rpm** puede utilizarse para instalar, actualizar, quitar y consultar paquetes RPM. Sin embargo, no resuelve dependencias de manera automática y todos los paquetes deben incluirse en una lista. Pueden utilizarse herramientas, como PackageKit y **yum**, así como aplicaciones front-end para **rpm** a fin de instalar paquetes individuales o *colecciones de paquetes* (a veces denominadas *grupos de paquetes*).

El comando **yum** permite buscar numerosos repositorios de paquetes y sus dependencias para que puedan instalarse de manera conjunta con la finalidad de atenuar los problemas de dependencia. El archivo de configuración principal de **yum** es `/etc/yum.conf` con archivos de configuración de repositorios adicionales ubicados en el directorio `/etc/yum.repos.d`. Los archivos de configuración de repositorio incluyen, como mínimo, una identificación de repositorio (en corchetes), un nombre y la ubicación de la URL del repositorio de paquetes. La URL puede apuntar a un directorio (archivo) local o recurso compartido de red remoto (`http`, `ftp`, etc.). Si la URL se pega en un navegador, los contenidos deben incluir los paquetes RPM, posiblemente en uno o más subdirectorios, y un directorio `repodata` con información sobre los paquetes disponibles.

El comando **yum** se utiliza para incluir en una lista repositorios, paquetes y grupos de paquetes:

```
[root@serverX ~]# yum repolist
```

```
Loaded plugins: langpacks
```

repo id	repo name	status
!rhel_dvd	Remote classroom copy of dvd	4,529

```
repolist: 4,529
```

```
[root@serverX ~]# yum list yum*
```

```
Loaded plugins: langpacks
```

```
Installed Packages
```

yum.noarch	3.4.3-118.2.el7	@anaconda/7.0
yum-langpacks.noarch	0.4.2-3.el7	@rhel_dvd
yum-metadata-parser.x86_64	1.1.4-10.el7	@anaconda/7.0
yum-rhn-plugin.noarch	2.0.1-4.el7	@rhel_dvd
yum-utils.noarch	1.1.31-24.el7	@rhel_dvd

```
Available Packages
```

```
yum-plugin-aliases.noarch      1.1.31-24.el7      rhel_dvd
yum-plugin-changelog.noarch    1.1.31-24.el7      rhel_dvd
yum-plugin-tmprepo.noarch      1.1.31-24.el7      rhel_dvd
yum-plugin-verify.noarch       1.1.31-24.el7      rhel_dvd
yum-plugin-versionlock.noarch  1.1.31-24.el7      rhel_dvd
[root@serverX ~]# yum list installed
Loaded plugins: langpacks
Installed Packages
GConf2.x86_64                  3.2.6-8.el7        @rhel_dvd
ModemManager.x86_64           1.1.0-6.git20130913.el7 @rhel_dvd
ModemManager-glib.x86_64      1.1.0-6.git20130913.el7 @rhel_dvd
...
[root@serverX ~]# yum grouplist
...
Installed groups:
  Base
  Desktop Debugging and Performance Tools
  Dial-up Networking Support
  Fonts
  Input Methods
...
```

## Referencias

Páginas de manual **yum**(8), **yum.conf**(5), **rpm**(8), **rpm2cpio**(8) y **rpmkeys**(8)

[Back](#)[Next](#)