



Examen 102 - Instalación de Linux y Gestión de Paquetes

102.5 Gestión de paquetes RPM y YUM

Teoría

En distribuciones como Red Hat Enterprise Linux (RHEL), CentOS Stream, Fedora, y Oracle Linux, el formato de paquete estándar es RPM (Red Hat Package Manager). Al igual que `.deb`, los archivos `.rpm` contienen el software y sus metadatos. La gestión de estos paquetes a gran escala y con resolución de dependencias se realiza tradicionalmente con YUM (Yellowdog Updater, Modified), o con su sucesor más moderno, DNF (Dandified Yum), que es el gestor de paquetes por defecto en las versiones recientes de Fedora, RHEL y CentOS Stream.

Conceptos Fundamentales:

1. **Paquete (`.rpm`):** Es un archivo que contiene el software, información de instalación, scripts y metadatos similares a los paquetes `.deb`. Los nombres de archivo RPM suelen seguir un formato estándar: `nombre-version-release.arquitectura.rpm` (ej: `httpd-2.4.37-62.el8.x86_64.rpm`).
2. **rpm:** Es la herramienta de bajo nivel para trabajar con archivos `.rpm` individuales o consultar la base de datos local de paquetes instalados (`/var/lib/rpm/`). Al igual que `dpkg`, *no* resuelve dependencias automáticamente durante la instalación de un archivo local.
 - `rpm -i <archivo.rpm>`: Instala un paquete desde un archivo `.rpm`.
 - `rpm -vhi <archivo.rpm>`: Instala con salida detallada (`v`), mostrando progreso con `#` (`h`), e imprime información de instalación (`i`).
 - `rpm -q <nombre_paquete>`: Consulta el estado de un paquete instalado (ej: `rpm -q httpd`).
 - `rpm -qa`: Lista *todos* los paquetes instalados (`query all`).
 - `rpm -ql <nombre_paquete>`: Lista los archivos instalados por un paquete.
 - `rpm -qf <ruta_archivo>`: Busca qué paquete instaló un archivo determinado (`query file`).
 - `rpm -e <nombre_paquete>`: Elimina un paquete.
 - `rpm -U <archivo.rpm>`: Actualiza un paquete (`upgrade`). Si no está instalado, lo instala.
 - `rpm -F <archivo.rpm>`: Actualiza un paquete (`freshen`). Solo actualiza si ya está instalado.
3. **YUM / DNF:** Son las herramientas de alto nivel (front-end) que utilizan `rpm` en segundo plano, pero añaden funcionalidades cruciales como la gestión de repositorios y la resolución automática de dependencias. **DNF es el sucesor recomendado de YUM.** La sintaxis de los comandos es muy similar entre ambos.

- **Configuración de Repositorios:** Los repositorios se definen en archivos `.repo` dentro del directorio `/etc/yum.repos.d/` (utilizado tanto por YUM como por DNF). Cada archivo `.repo` define uno o más repositorios con detalles como nombre, URL base, si está habilitado, opciones de verificación de clave GPG, etc. El archivo principal de configuración global puede ser `/etc/yum.conf` (YUM) o `/etc/dnf/dnf.conf` (DNF).
- **Comandos Comunes (válidos para yum y dnf, se recomienda dnf):**
 - `[sudo] dnf check-update`: Comprueba si hay actualizaciones disponibles en los repositorios. (Equivalente a `apt update` en el sentido de sincronizar la información *después* de que los repositorios ya estén definidos).
 - `[sudo] dnf upgrade`: Actualiza todos los paquetes instalados a sus versiones más recientes.
 - `[sudo] dnf install <nombre_paquete>`: Instala un paquete (y todas sus dependencias) desde los repositorios.
 - `[sudo] dnf remove <nombre_paquete>`: Elimina un paquete (y, opcionalmente, dependencias que ya no son necesarias, aunque DNF lo hace mejor que YUM en esto).
 - `dnf search <término>`: Busca paquetes que coincidan con el término.
 - `dnf info <nombre_paquete>`: Muestra información detallada sobre un paquete (disponible o instalado).
 - `dnf list installed [<patrón>]`: Lista los paquetes instalados (similar a `rpm -qa`, pero puede interactuar con repositorios).
 - `dnf autoremove`: Elimina paquetes que se instalaron como dependencias pero ya no son necesarios para ningún paquete instalado explícitamente.
 - `dnf history`: Muestra un historial de transacciones de DNF/YUM.

Comparación APT vs. RPM/YUM/DNF:

Son sistemas diferentes con formatos de paquete distintos (`.deb` vs `.rpm`) y herramientas de gestión de bajo nivel (`dpkg` vs `rpm`) y alto nivel (`apt` vs `yum/dnf`). Sin embargo, sus funcionalidades principales (instalación, actualización, eliminación, gestión de dependencias, uso de repositorios) son conceptualmente muy similares. La elección entre uno u otro depende de la distribución Linux que se utilice.