Packages & Software Management (Core)

Linux Commands Course · Section 13

Goal

Learn how to install, update, and manage software in Linux using package managers.

You'll understand what packages are, how repositories work, and how to use the major tools — especially apt on Debian/Ubuntu systems.

What Are Packages?

A package is a compressed bundle that contains:

- Program files (binaries, libraries, icons)
- Configuration files
- Metadata (version, dependencies, maintainer info)

Instead of manually copying files, the package manager handles installation, updates, and removal automatically.

Where Packages Come From — Repositories

Linux distributions host packages on remote repositories (repos) — organized servers containing signed software.

Each system has a list of repositories stored in config files, such as:

- /etc/apt/sources.list (Debian/Ubuntu)
- /etc/yum.repos.d/ (RHEL/Fedora)
- /etc/pacman.conf (Arch)
- /etc/zypp/repos.d/ (openSUSE)

The package manager connects to these servers to download and verify packages.

APT (Advanced Package Tool) — Debian/Ubuntu

APT is the package manager used by **Debian**, **Ubuntu**, and their derivatives (Mint, Kali, etc.).

Updating Package Information

Before installing anything, update your local list of available software:

sudo apt update

This syncs your system with the repository metadata — names, versions, and dependencies.

Then upgrade installed software:

sudo apt upgrade

- apt update → refreshes the list
- apt upgrade → installs newer versions of already-installed packages

To upgrade all packages and remove obsolete ones:

sudo apt full-upgrade

Installing Packages

Install	one	or	multiple	packages:	

sudo apt install curl vim git

APT automatically downloads dependencies and installs them.

Install a specific version:

sudo apt install nginx=1.18.0-0ubuntu1

Removing Packages

Remove a package but keep its config files:		
	sudo apt remove nginx	
Remove a package and its configs:		
	sudo apt purge nginx	
Clean up unnecessary packages and cache:		
	sudo apt autoremove sudo apt clean	

Inspecting Packages



Installing Local .deb Files - dpkg

Install a .deb file manually (downloaded from a website):

sudo dpkg -i package.deb

If dependencies are missing, fix them with:

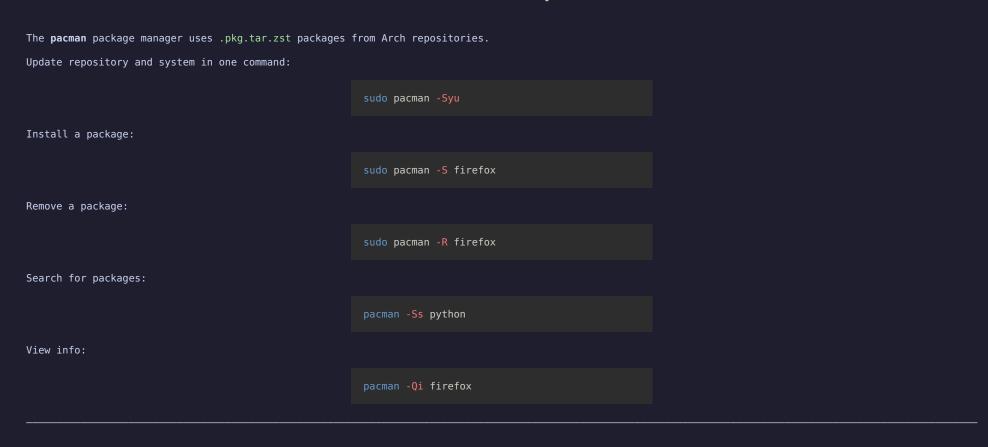
sudo apt -f install

This tells APT to install the required packages automatically.

RHEL/Fedora — dnf and rpm

```
dnf (successor to yum) is used on RHEL, Fedora, and CentOS.
Install software:
                                                      sudo dnf install nginx
Remove software:
                                                      sudo dnf remove nginx
Update all packages:
                                                      sudo dnf update
Query package info:
                                                      dnf info nginx
Manual package management via RPM:
                                                      sudo rpm -qi nginx # info
                                                      sudo rpm -ql nginx # list files
                                                      sudo rpm -ivh file.rpm # install
                                                      sudo rpm -e package # remove
```

Arch Linux - pacman



openSUSE - zypper

zypper is the package tool for openSUSE and SLE system	S.				
Refresh repositories:					
	sudo zypper refresh				
Install packages:					
	sudo zypper in vim				
Remove packages:					
	sudo zypper rm vim				
Update system:					
	sudo zypper up				
Show package info:					
	zypper info vim				

Universal Package Systems

Some distributions support universal formats — portabl	e across distros.
Snap	
Developed by Canonical, runs sandboxed applications.	
List installed snaps:	
	snap list
Install a snap package:	
	sudo snap install codeclassic
Remove a snap:	
	sudo snap remove code
Flatpak	
Another universal, sandboxed format.	
Install a Flatpak application:	
	flatpak install flathub org.gimp.GIMP
Run a Flatpak app:	
	flatpak run org.gimp.GIMP
List installed Flatpaks:	
	flatpak list

Comparing Package Managers

Distro	Tool	Install Example	Notes
Debian/Ubuntu	apt	apt install nginx	Most common; uses .deb
RHEL/Fedora	dnf	dnf install nginx	Uses .rpm
Arch	pacman	pacman -S nginx	Very fast, rolling updates
openSUSE	zypper	zypper in nginx	Enterprise-grade
Universal	snap, flatpak	Cross-platform apps	Great for desktop software

Tips & Best Practices

- Always run sudo apt update before installing.
- Prefer repos over manual .deb downloads ensures security & updates.
 Avoid mixing package types (e.g., .deb + Snap + Flatpak) unless needed.
- Periodically clean unused packages with autoremove.
- Review installed packages with apt list --installed.

Recap

- APT update, install, remove, purge, inspect packages
 Repositories centralized sources of verified software
- dpkg for manual .deb installs
 dnf / rpm, pacman, zypper alternatives for other distros
 snap / flatpak universal sandboxed packages

Mastering package management makes system maintenance fast, secure, and reliable.

Practice

- Run sudo apt update && sudo apt upgrade.
 Install and remove curl using APT.
 Install a .deb package manually and fix dependencies.
 List all installed packages containing "python".
 Query info for an installed package with apt show.
 Try installing and launching a Snap or Flatpak application.

Next Up

Disks & Filesystems (Core) — managing partitions, mounts, and storage usage.