

Managing Software with APT, DNF, and Snap

Software management on modern Linux distributions

Speakers:







- 22120368 - Phan Thanh Tiến
- 22120375 - Lưu Thái Toàn
- 22120383 - Nguyễn Đăng Trí

Objectives

- Understand the role of package managers
- Add and remove a repository
- Search / install / remove packages
- Compare APT, DNF, and Snap
- Demo

Background: Package managers

- A tool for **discovery, install, upgrade, and removal** of system package (including apps and its libraries to run). *Not limited to Linux systems!*

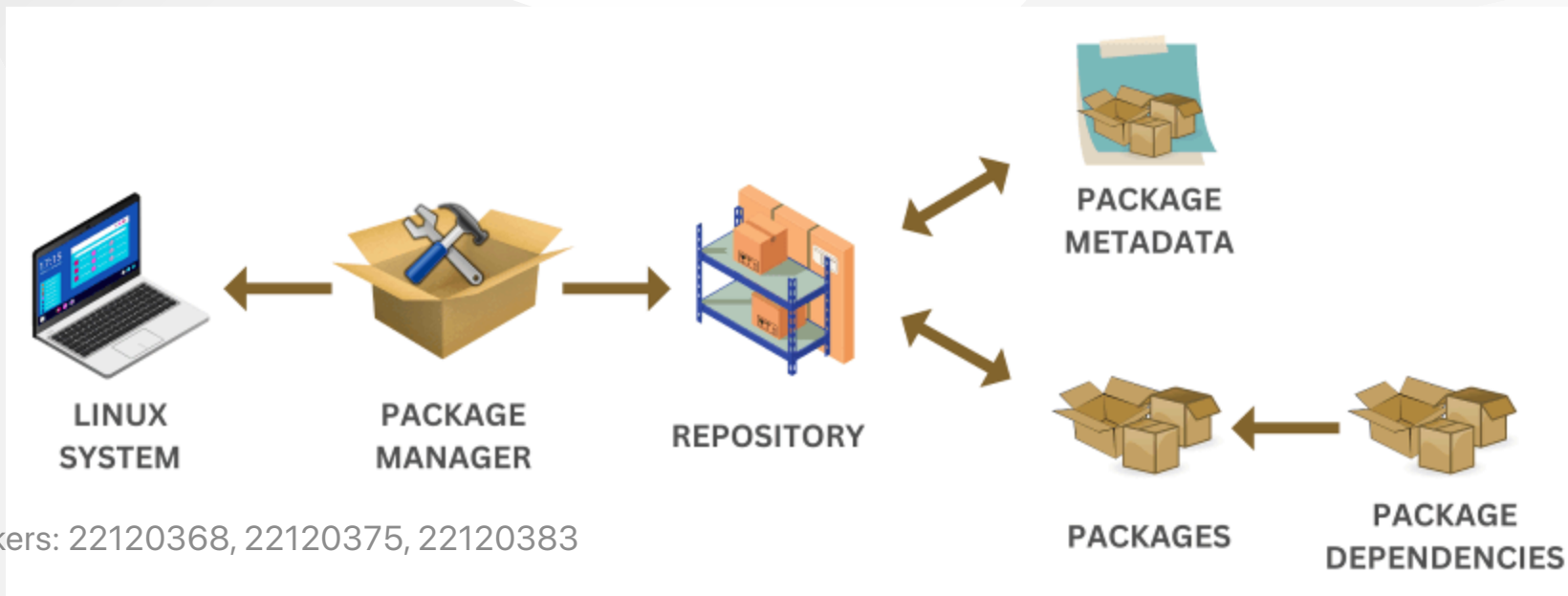
Windows	Linux	MacOS
	  fedora  	
Chocolatey	APT / DNF (Debian / RPM) • Snap • Flatpak (app sandboxing)	Homebrew

- They perform dependency resolution, manage metadata and caches, and apply updates
- They integrate with the OS packaging format (.deb / .rpm) and can influence system state (services, configs)
- Important for security (timely patches), reproducibility, and disk/resource sharing

- Types of package managers:
 - Imperative: modify system state directly (e.g., APT, DNF)
 - Application package manager: bundle app + dependencies, isolated from system (e.g., Snap, Flatpak)
 - Declarative: define desired state, system converges to it (e.g., Nix Package Manager)

Background: Repositories and packages

- Repository: a remote source (URL) that exposes package metadata and files
- Packages: versioned bundles (binary + metadata) built for the distro format
- Repo metadata (indexes) enable search, dependency resolution, and fast installs via caching
- Trust model: repositories are usually signed with GPG keys — verify before adding



Imperative package formats

`apt` and `dnf`

Why "imperative"?

- The system state will change immediately after you run the command.
- "System state" includes installed packages, configuration files, services, libraries, and overall system behavior.
- After running an install/upgrade/remove command, the system will reflect those changes right away. It may require restarting services or logging out/in to fully apply changes.

When install a package...

- Signature & integrity checks
- Resolve dependencies
- Install dependencies and the package
- Triggers and post-install integration

APT (Debian / Ubuntu)

- Frontends: apt, apt-get, apt-cache
- Common commands:
 - Update metadata: `sudo apt update`
 - Search: `apt search <name>` or `apt-cache search <name>`
 - Install: `sudo apt install <package>`
 - Remove: `sudo apt remove <package>` (keep config) or `sudo apt purge <package>` (remove config)
- Repo files: `/etc/apt/sources.list` and `/etc/apt/sources.list.d/`

Repository structure (.list file)

```
deb [repository_url] [distribution] [component]
```

- **distribution** : Specified the distribution name of the Debian (e.g., stable, buster, focal) or Ubuntu (e.g., focal, jammy)
- **component** : Define the component which can be main, contrib, and non-free

Repository structure (.sources file)

- More declarative. Being recommended recently.
- Structure:

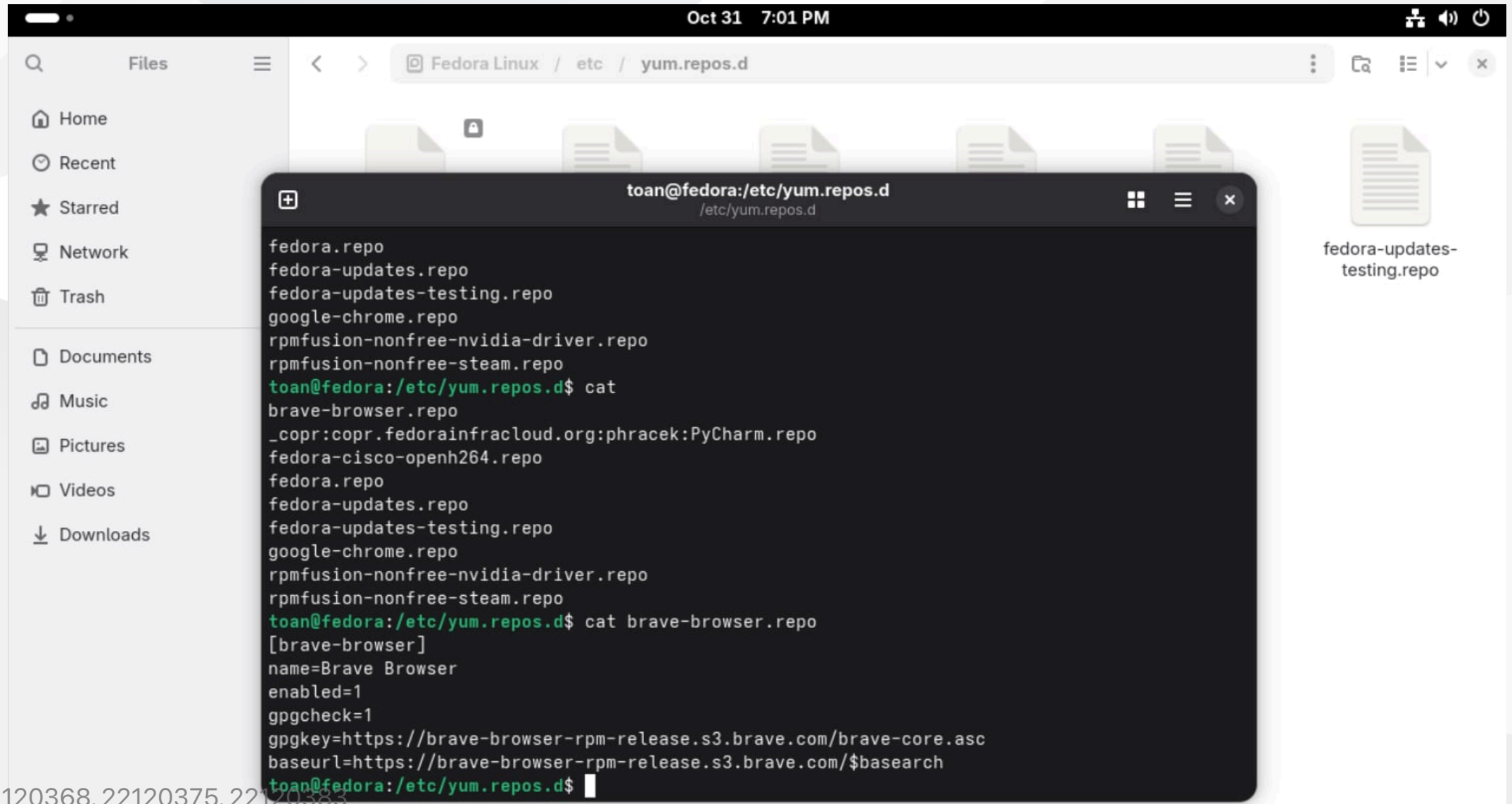
```
Types: deb
URIs: [repository_url]
Suites: [distribution: stable, buster, focal...]
Architectures: amd64 | i386 | arm64 | all
Components: [component: main, contrib, non-free]
Signed-By: [path_to_GPG_key]
```

```
❯ cat /etc/apt/sources.list.d/kubernetes.list.save
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.33/deb/ /
❯ cat /etc/apt/sources.list.d/kubernetes.sources
Types: deb
URIs: https://pkgs.k8s.io/core:/stable:/v1.33/deb/
Suites:
Components:
Signed-By: /etc/apt/keyrings/kubernetes-apt-keyring.gpg
Enabled: no
```

DNF (Fedora / RHEL / CentOS / AlmaLinux)

- Successor to yum on many RPM-based distros
- Common commands:
 - Update metadata: `sudo dnf makecache` or `sudo dnf check-update`
 - Search: `dnf search <name>`
 - Install: `sudo dnf install <package>`
 - Remove: `sudo dnf remove <package>`
- Compared to apt: <https://docs.fedoraproject.org/en-US/quick-docs/dnf-vs-apt/>

- Repo files: `/etc/yum.repos.d/*.repo`



Cross platform package managers

Snap and Flatpak

Snap packages

- Bundle the app and most of its dependencies into a single compressed file (squashfs).
- Some snaps use **shared content snaps** (like GNOME or KDE runtimes) to avoid duplicating large libraries.
- Managed by the `snapped` service, which handles installing, updating, and running snaps.

- Cross-distro compatibility:
 - Snaps do not rely on the host's package manager or libraries (no .deb/.rpm needed).
 - As long as `snapd` is installed, snaps work the same way on any Linux distribution.
- Sandboxing:
 - Snaps run in a confined environment, limiting access to system resources for security.
 - Permissions are managed via interfaces that can be connected or disconnected.

Snap (Canonical)

- Find: `snap find <name>`
- Install: `sudo snap install <snap-name>`
- Remove: `sudo snap remove <snap-name>`
- List installed: `snap list`

<https://snapcraft.io/docs/snap-howto>

- *Snap is a centralized app store, managed by Canonical. So there are no way to add/remove repository like APT or DNF.*

Comparison: APT | DNF | Snap

APT	DNF	Snap
Package format: .deb	Package format: .rpm	Package format: snap bundle
Repo files: /etc/apt/sources.list(.d)	Repo files: /etc/yum.repos.d/*.repo	Managed by snapd (no distro repo files)
Strong dependency resolution (dpkg backend)	Dependency resolution with rich metadata, plugins	Bundles dependencies, isolated runtime (larger size)
System-focused packages, integrates with system services	System-focused packages, plugin ecosystem	App-focused, sandboxed, transactional installs

APT (Debian/Ubuntu)	DNF (Fedora/RHEL/AlmaLinux)	Snap (cross-distro)
Typical use: system packages, servers, libraries	Typical use: system packages, enterprise RHEL ecosystems	Typical use: desktop and some server apps for cross-distro delivery
Pros: mature, fast, small packages	Pros: modern metadata, modular repos	Pros: cross-distro, sandboxed, easy packaging
Cons: tied to Debian ecosystem	Cons: tied to RPM ecosystem	Cons: larger disk usage, sometimes slower start, requires snapd

Demo

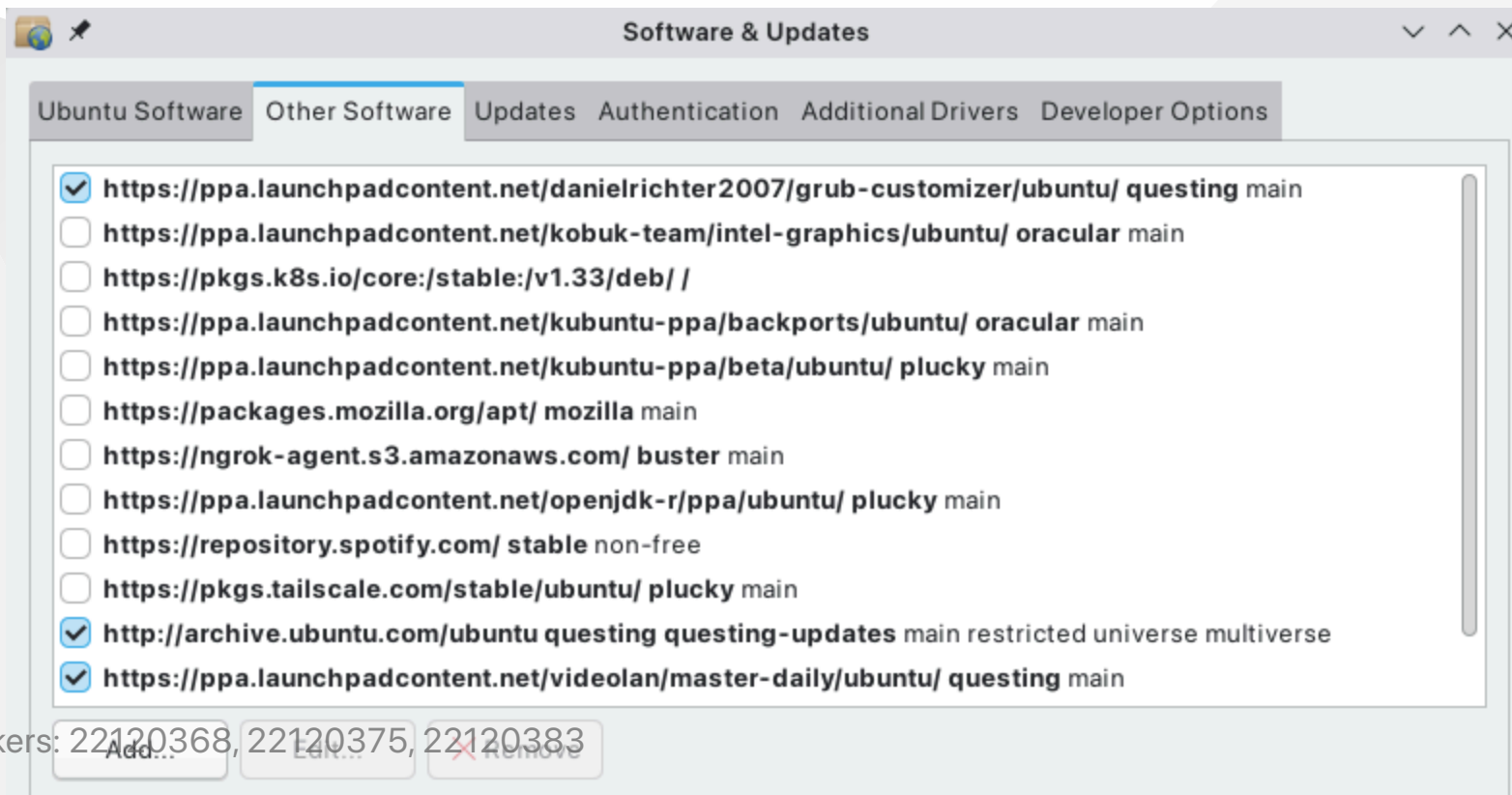
Install a package that not exists in default repositories by adding a new repository

https://www.sublimetext.com/docs/linux_repositories.html

Debian/Ubuntu

Add a new repository

- Edit `/etc/apt/sources.list`
- Use GUI software center (e.g., Ubuntu Software)



```
# add GPG key
wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | \
  sudo tee /etc/apt/keyrings/sublimehq-pub.asc > /dev/null

# .list file
echo "deb [signed-by=/etc/apt/keyrings/sublimehq-pub.asc] https://download.sublimetext.com/ apt/stable/" \
| sudo tee /etc/apt/sources.list.d/sublime-text.list >/dev/null

# .source file
echo -e \
"Types: deb
URIs: https://download.sublimetext.com/
Suites: apt/stable/
Architectures: amd64
Components: main
Signed-By: /etc/apt/keyrings/sublimehq-pub.asc" | sudo tee /etc/apt/sources.list.d/sublime-text.source

# apt-add-repository
sudo apt-add-repository "deb [signed-by=/etc/apt/keyrings/sublimehq-pub.asc] \
https://download.sublimetext.com/ apt/stable/"
```


Debian/Ubuntu

Installing a software from added repositories

```
sudo apt update
apt search sublime-text | head -n 20
sudo apt install sublime-text
apt list --installed | grep sublime-text || true
sudo apt remove sublime-text

sudo add-apt-repository -r "deb https://download.sublimetext.com/ apt/stable/"
```

- GUI: Any software center (e.g., Ubuntu Software/GNOME Software/KDE Discover).
- Or Synaptic Package Manager.

CentOS/RHEL/AlmaLinux

- Adding/Disabling/Removing repo:

<https://gist.github.com/aelkz/0dc6864cd7f3665a2780b2a111ad1a49>

```
sudo rpm -v --import \  
https://download.sublimetext.com/sublimehq-rpm-pub.gpg  
sudo dnf config-manager --add-repo \  
https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo
```

```
sudo dnf makecache  
sudo dnf install -y sublime-text  
dnf list installed | grep sublime-text || true  
sudo dnf remove -y sublime-text  
sudo dnf config-manager --remove-repo \  
https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo
```

Snap

```
sudo apt install -y snapd
snap find hello-world | head -n 10
sudo snap install hello-world
snap list | head -n 20
sudo snap remove hello-world
```

Trends & alternatives

- App distribution alternatives: Flatpak and AppImage (app sandboxing, desktop apps)
- Functional/declarative package managers: Nix / NixOS (focus on reproducibility, rollbacks)
- Containers change distribution of applications, but package managers remain important for system maintenance and shared libraries

References

- `man apt`, `man apt-get`, `man dnf`, `man snap`
- <https://snapcraft.io/docs>
- Flatpak docs: <https://flatpak.org>
- Nix/NixOS introduction: <https://nixos.org>

Thank you

Questions?

