

Linux Operating System and Applications

Managing Software with APT, DNF, and Snap

Members:

22127089 | Tran Le Bao Duy

22127455 | Vu Thanh Viet

I. Giới thiệu chung

Mục tiêu: Tìm hiểu và sử dụng ba công cụ phổ biến trong quản lý phần mềm trên Linux: APT, DNF và Snap.

Tầm quan trọng: Quản lý phần mềm là kỹ năng cốt lõi cho người dùng Linux – từ cài đặt, cập nhật đến gỡ bỏ phần mềm.

Khái niệm Repository (Repo)

Repository (kho lưu trữ phần mềm) là một tập hợp các gói phần mềm được lưu trữ trên một máy chủ và được quản lý tập trung.

- Repo bao gồm:
 - Các gói phần mềm (.deb, .rpm, .snap,...)
 - Metadata: mô tả gói, phụ thuộc, phiên bản, kích thước,...
 - Các tệp tin ký số: đảm bảo an toàn và xác thực nguồn gốc
- Mỗi công cụ quản lý gói (APT, DNF, Snap) sử dụng repo để tìm, tải, và cập nhật phần mềm.

Vai trò của Repository trong hệ thống quản lý phần mềm

- Cung cấp phần mềm chính thức
- Cập nhật phần mềm dễ dàng
- Giải quyết phụ thuộc
- Đảm bảo độ tin cậy và an toàn
- Tự động hóa quản lý phần mềm

Khái niệm Repository (Repo)

Có những loại repository nào?

- Chính thức (official): Do nhà phát triển hệ điều hành cung cấp (Ubuntu, Fedora,...) - an toàn, ổn định
- Bên thứ ba (third-party): Do nhà phát triển độc lập tạo ra (PPA, EPEL, Google repo, Docker repo,...)
- Repo thử nghiệm (testing/unstable): Dành cho người dùng muốn thử phần mềm mới, có thể ít ổn định
- Repo nội bộ (local): Tự thiết lập trong tổ chức để triển khai phần mềm trong mạng nội bộ

Repo hoạt động như thế nào?

Khi gõ lệnh như:

sudo apt install vlc

Hệ thống sẽ:

1. Đọc danh sách repo trong máy (/etc/apt/sources.list hoặc /etc/yum.repos.d/)
2. Tìm thông tin gói vlc trong metadata đã tải trước đó (apt update / dnf check-update)
3. Tải gói .deb hoặc .rpm và các phụ thuộc cần thiết từ repo.
4. Cài đặt và cấu hình tự động.

Vì vậy: nếu repo bị lỗi, không hợp lệ, hoặc gói không có trong repo → không thể cài được.

II. Tổng quan về các công cụ

Công cụ	Hệ điều hành chính	Hệ quản lý gói	Ngôn ngữ chính
APT	Debian/Ubuntu	.deb	Bash
DNF	Fedora/RHEL/CentOS	.rpm	Python
Snap	Đa nền tảng	Snap packages	Go

III. APT (Advanced Package Tool)

1. Giới thiệu về APT

APT (Advanced Package Tool) là công cụ quản lý gói phần mềm phổ biến trên các hệ điều hành Debian-based, ví dụ:

Ubuntu, Linux Mint, Kali Linux, Elementary OS, Pop!_OS,...

APT hoạt động như một giao diện dòng lệnh cho hệ thống quản lý gói .deb.

2. Kiến trúc hoạt động của APT

APT tương tác với 3 thành phần chính:

Sources list: Danh sách các kho lưu trữ phần mềm (/etc/apt/sources.list) chứa đường dẫn đến các server hoặc mirror.

Package database: Cơ sở dữ liệu lưu thông tin tất cả các gói có thể cài và đã cài.

DPKG: Trình quản lý gói cấp thấp, chịu trách nhiệm cài đặt/gỡ bỏ .deb.

III. APT (Advanced Package Tool)

3. Các thao tác cơ bản với APT

Cập nhật và nâng cấp hệ thống:

sudo apt update # Cập nhật danh sách gói phần mềm mới từ kho

sudo apt upgrade # Nâng cấp tất cả phần mềm đã cài lên phiên bản mới nhất

Cài đặt phần mềm:

sudo apt install firefox # Cài đặt Firefox

sudo apt install nginx curl # Cài đặt nhiều gói cùng lúc

Gỡ bỏ phần mềm:

sudo apt remove nginx # Gỡ phần mềm nhưng giữ lại file cấu hình

sudo apt purge nginx # Gỡ hoàn toàn, bao gồm file cấu hình

Tìm kiếm và xem thông tin:

apt search vlc # Tìm kiếm gói theo tên

apt show vlc # Xem thông tin chi tiết của gói

Dọn dẹp hệ thống:

sudo apt autoremove # Gỡ các gói không còn cần thiết

sudo apt clean # Xóa cache đã tải

III. APT (Advanced Package Tool)

4. Repository của APT

Cấu trúc repo APT:

- Chứa các gói phần mềm .deb và file metadata.
- Chia thành các thành phần:
 - main: phần mềm mã nguồn mở được Ubuntu/Debian hỗ trợ chính thức
 - universe: phần mềm mã nguồn mở do cộng đồng đóng góp
 - restricted: driver hoặc phần mềm độc quyền
 - multiverse: phần mềm không hoàn toàn tự do (license không mở)

III. APT (Advanced Package Tool)

4. Repository của APT

Vị trí cấu hình:

File chính:

- /etc/apt/sources.list.d/ubuntu.sources

Thư mục chứa repo bổ sung:

- /etc/apt/sources.list.d/

Quản lý repo:

-Thêm repo:

- sudo add-apt-repository ppa:<username>/<repo>

-Cập nhật sau khi thêm:

- sudo apt update

IV. DNF (Dandified YUM)

1. Giới thiệu về DNF

DNF (viết tắt của “Dandified YUM”) là trình quản lý gói thế hệ mới, thay thế cho YUM (Yellowdog Updater Modified).

Xuất hiện đầu tiên trong Fedora 18 và chính thức thay thế YUM từ Fedora 22.

DNF dùng để cài đặt, cập nhật, gỡ bỏ phần mềm với các gói .rpm.

2. Kiến trúc hoạt động của DNF

DNF xử lý tương tự APT, nhưng sử dụng định dạng gói .rpm. Cấu trúc bao gồm:

Repository: Được định nghĩa trong thư mục /etc/yum.repos.d/ (đuôi .repo).

Metadata cache: DNF sử dụng SQLite để lưu cache thay vì XML như YUM → nhanh và ít tốn RAM.

Dependency resolver: DNF sử dụng libolv để xử lý phụ thuộc thông minh, tránh xung đột.

IV. DNF (Dandified YUM)

3. Các thao tác cơ bản với DNF

Cập nhật hệ thống:

```
sudo dnf check-update      # Kiểm tra cập nhật  
sudo dnf upgrade          # Nâng cấp tất cả gói
```

Cài đặt phần mềm:

```
sudo dnf install firefox    # Cài đặt Firefox  
sudo dnf install nginx curl # Cài đặt nhiều gói
```

Gỡ bỏ phần mềm:

```
sudo dnf remove nginx      # Gỡ bỏ gói
```

Tìm kiếm và thông tin:

```
dnf search vlc            # Tìm kiếm gói  
dnf info vlc             # Xem thông tin gói
```

Dọn dẹp hệ thống:

```
sudo dnf autoremove        # Gỡ các gói không còn cần  
sudo dnf clean all         # Xóa cache và metadata
```

IV. DNF (Dandified YUM)

4. Repository của DNF

Cấu trúc repo DNF:

- Chứa các gói phần mềm .rpm và metadata ở định dạng SQLite hoặc XML.
- Các repo phổ biến:
 - fedora, updates, updates-testing
 - epel (Extra Packages for Enterprise Linux)

Vị trí cấu hình:

Repo nằm trong các file .repo tại thư mục:

- /etc/yum.repos.d/

IV. DNF (Dandified YUM)

4. Repository của DNF

Mỗi file có định dạng [fedora]:

- name=Fedora \$releasever - \$basearch
- baseurl=https://download.fedoraproject.org/pub/fedora/linux/releases/\$releasever/Everything/\$basearch/os/
- enabled=1
- gpgcheck=1

Quản lý repo:

Bật/tắt repo:

- sudo dnf --enablerepo=epel install <package>
- sudo dnf config-manager --set-disabled=fedora
- Thêm repo ngoài (ví dụ EPEL):
- sudo dnf install epel-release

V. Snap

1. Giới thiệu về Snap

Snap là định dạng gói phần mềm được phát triển bởi Canonical (công ty đứng sau Ubuntu).

Mục tiêu: giải quyết các vấn đề về phụ thuộc thư viện, tính di động, và bảo mật bằng cách đóng gói phần mềm theo kiểu "**tự đủ**" (**self-contained**).

Các gói Snap có thể chạy trên nhiều bản phân phối Linux khác nhau như:

Ubuntu, Fedora, Arch, Debian, Manjaro,...

"Tự đủ" (self-contained**) có nghĩa là:**

Gói phần mềm tự mang theo đầy đủ tất cả các thành phần cần thiết để chạy, không phụ thuộc vào thư viện hệ thống.

Cụ thể hơn:

- Khi bạn cài một gói Snap, nó không cần dùng thư viện có sẵn trong hệ điều hành (như thư viện libxyz ở hệ thống).
- Thay vào đó, Snap gói luôn các thư viện đó bên trong .snap file → đảm bảo chạy được trên mọi hệ thống Linux, bất kể distro.

V. Snap

2. Cơ chế hoạt động của Snap

a. Kiến trúc Snap

- Mỗi gói Snap chứa toàn bộ mã thực thi + thư viện cần thiết, gần như là một container nhỏ.
- Snap sử dụng Snapd - một daemon chạy nền giúp quản lý và cung cấp runtime cho các gói Snap.

b. Các thành phần chính:

- snap: CLI dùng để tương tác với Snap system.
- snapd: Dịch vụ nền quản lý snap và sandbox.
- Snap Store: Kho ứng dụng chính thức tương tự như App Store.

V. Snap

3. Các thao tác cơ bản với SNAP

Cài đặt phần mềm:

sudo snap install <package> # Cài đặt gói từ Snap Store

sudo snap install <package> --classic # Cài đặt chế độ classic (cho phần mềm cần quyền đầy đủ, như VS Code)

Gỡ bỏ phần mềm:

sudo snap remove <package> # Gỡ bỏ gói Snap

Tìm kiếm và thông tin:

sudo snap find <tên> # Tìm kiếm gói

sudo snap info <gói> # Xem thông tin chi tiết

Liệt kê và quản lý:

sudo snap list # Liệt kê tất cả gói Snap đã cài

sudo snap refresh # Cập nhật tất cả gói Snap

V. Snap

4. Repository của Snap- Snap Store:

Cấu trúc repo Snap:

- Không giống APT/DNF, Snap sử dụng Snap Store như một kho tập trung toàn cầu, không cần cấu hình thủ công.
- Các gói Snap là container độc lập và chạy sandboxed.

Quản lý repo:

- Snap tự động kết nối với Snap Store qua dịch vụ snapd.
- Người dùng không cần chỉnh sửa file repo như APT hay DNF.
- Snap không dùng file kiểu sources.list hay .repo, mà mọi thao tác đều dùng lệnh Snap.

V. Snap

4. Repository của Snap- Snap Store:

Ví dụ tìm và cài:

sudo snap find vlc

sudo snap install vlc

Kiểm tra Snap Store trạng thái:

sudo snap info vlc

V. Snap

5. Ưu – Nhược điểm của Snap

Ưu điểm:

- Tương thích đa nền tảng Linux.
- Đóng gói đầy đủ: không cần lo về thư viện phụ thuộc.
- Cập nhật tự động và rollback (quay lại bản cũ nếu lỗi).
- Sandboxing tăng cường bảo mật.

Nhược điểm:

- Dung lượng lớn hơn gói truyền thống (do đóng gói nhiều thư viện).
- Khởi động ứng dụng chậm hơn so với gói gốc.
- Một số người dùng không thích do Snapd chạy nền và chiếm tài nguyên.
- Cấu hình hệ thống và quyền truy cập không linh hoạt bằng .deb/.rpm.

VI. Reference

APT (Advanced Package Tool).

DNF (Dandified YUM).

Snap

Repository_(Repo).

Thank You!