Công ty An ninh mạng Viettel

Báo cáo về Package Management

Sinh viên: Nguyễn Đan Trường

*MỤC LỤC*

[*1. Package 3*](#_Toc151732641)

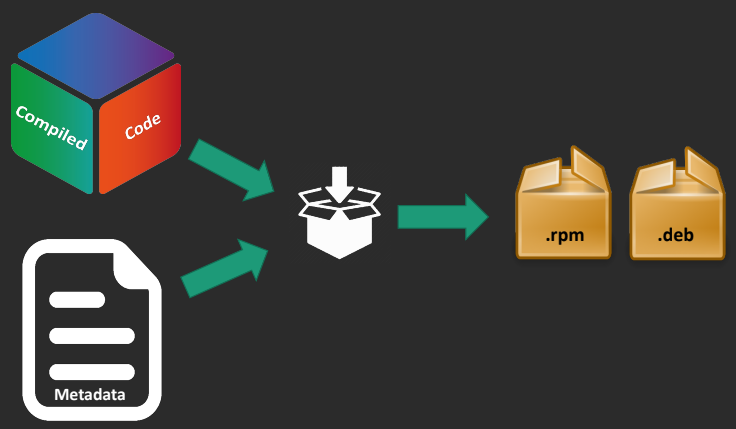
[*2. YUM và APT 14*](#_Toc151732642)

[*3. Cấu hình repo 15*](#_Toc151732643)

[*4. Add/modify/remove repo 15*](#_Toc151732644)

## Package

Trong linux, Package là một tập hợp các tệp tin được đóng gói cùng nhau để cài đặt một ứng dụng, thư viện hoặc một thành phần hệ thống cụ thể. Các gói thường chứa mã nguồn, tệp tin cấu hình, tài nguyên và thông tin cần thiết để cài đặt và chạy một ứng dụng.



Đối với từng distro cách đặt tên package cũng sẽ có phần khác nhau:

* Redhat:

***<Package name>-<Version>-<Release>.<Architecture>.rpm***

docker-1.13.1-94.gitb2f74b2.el7.centos.x86\_64.rpm

emacs-24.3-22.el7.x86\_64.rpm

openssh-7.4p1-16.el7.x86\_64.rpm

zsh-5.0.2-31.el7.x86\_64.rpm

* Debian:

***<Package name>\_<Version>-<Release>\_<Architecture>.deb***

docker\_1.5-1build1\_amd64.deb

emacs\_47.0\_all.deb

openssh-client\_1%3a7.6p1-4ubuntu0.3\_amd64.deb

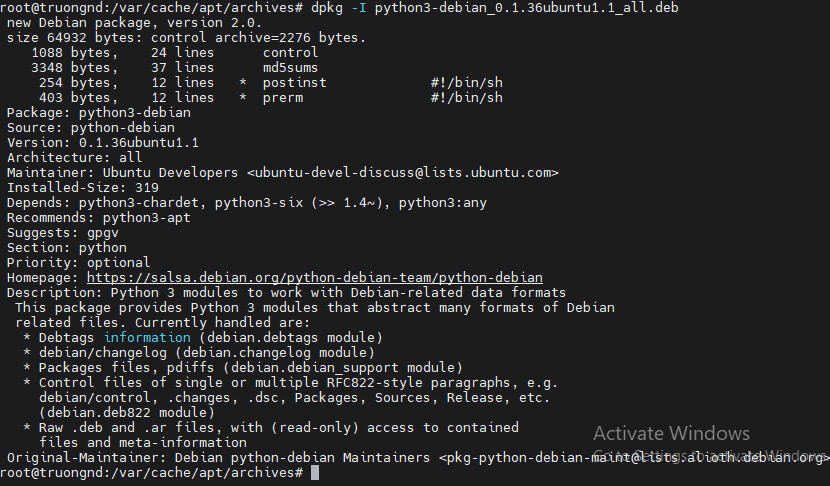
vim\_2%3a8.0.1453-1ubuntu1\_amd64.deb

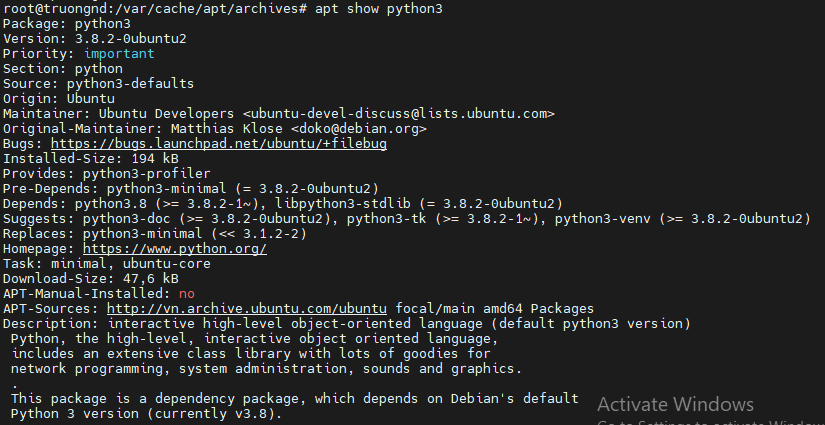
zsh\_5.4.2-3ubuntu3.1\_amd64.deb

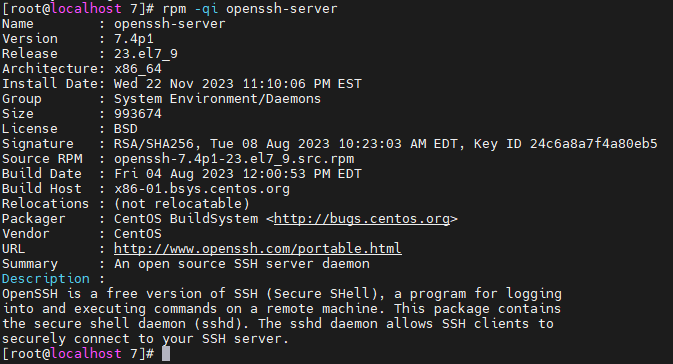
Bên trong một package sẽ bao gồm hai thành phần chính đó là: metadata và mã nguồn chương trình.

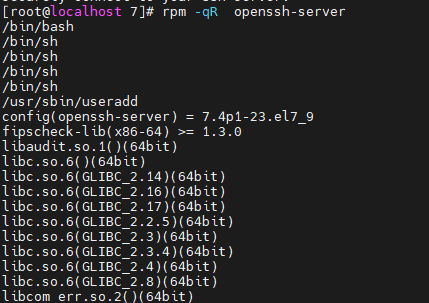
**Metadata**

* **Package Information**: Bao gồm tên gói, phiên bản, mô tả, tác giả, giấy phép sử dụng, và các thông tin liên quan khác**.**
* **Dependencies**: Các gói phụ thuộc.
* **Chữ ký số**: Đảm bảo tính toàn vẹ và nguồn gốc của gói.
* **Permissions và Ownership**: Thông tin về quyền truy cập và quyền sở hữu tệp tin trong package.



****

****

****

**Mã nguồn (code)**

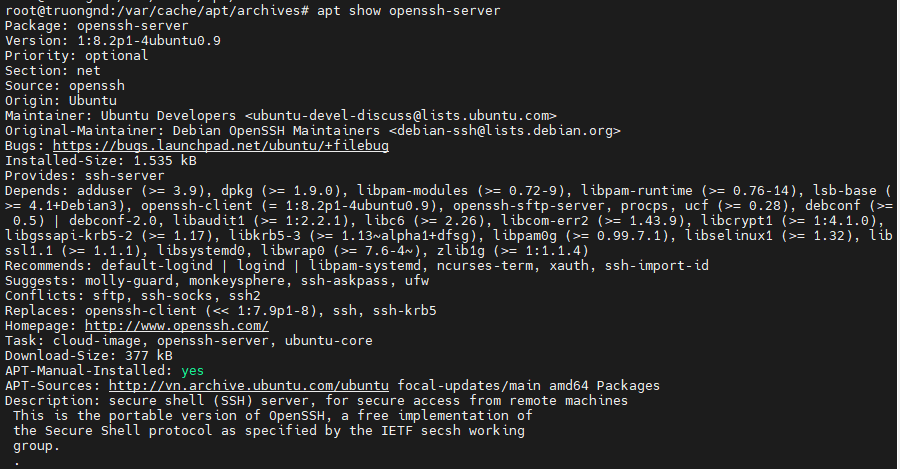
* **Thư mục nguồn**: Chứa mã nguồn gốc của ứng dụng.
* **Tệp tin cấu hình**: Nếu có, chứa các tệp tin cấu hình mà hệ thống có thể chạy để khởi động ứng dụng.
* **Tệp thực thi**: Bao gồm các tệp tin thực thi mà hệ thống có thể chạy để khởi động ứng dụng.
* **Tệp thư viện**: Các thư viện cần thiết để ứng dụng hoạt đông.
* **Tệp tài nguyên**: Các tệp tin khác như hình ảnh, âm thanh …
* **Tệp tin tài liệu**: Các tệp tin tài liệu về cách cài đặt, sử dụng và cấu hình ứng dụng.

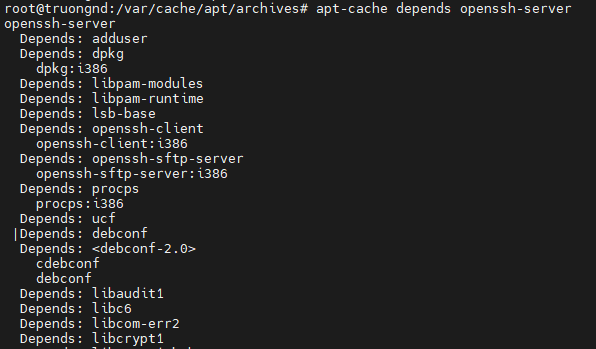
**Package dependency**

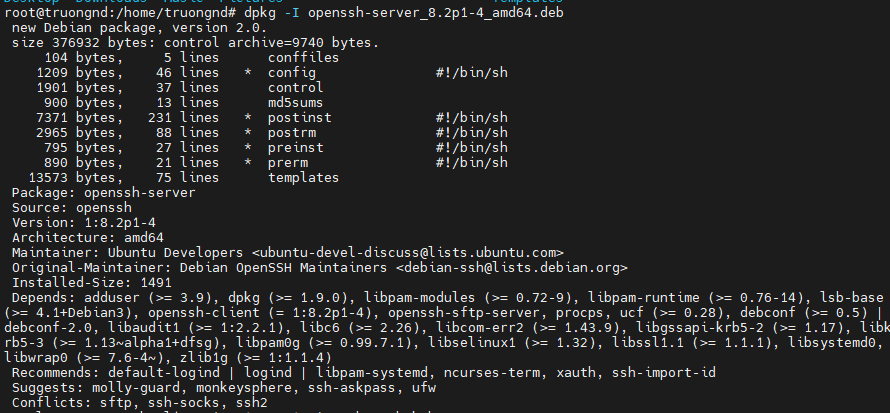
Đôi khi cài đặt một gói phần mềm trong Linux cần phải cài đặt các gói khác trước để hoạt động bình thường. Những gói bổ sung này được gói là package dependenies.

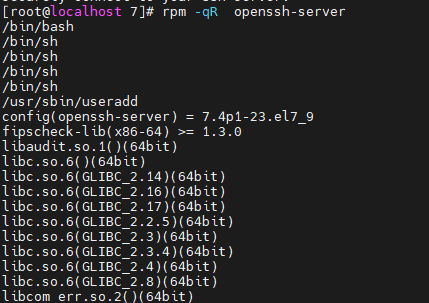
Nếu các package dependencies không được cài đặt trên hệ thống, nó thường sẽ được tự động cài đặt cùng với gói chính.

Kiểm tra dependency*:*

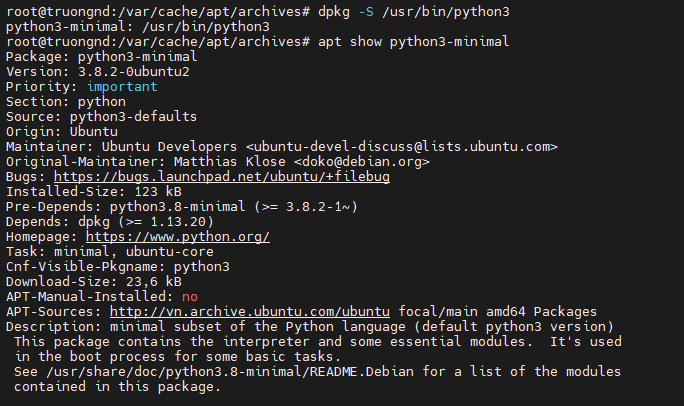


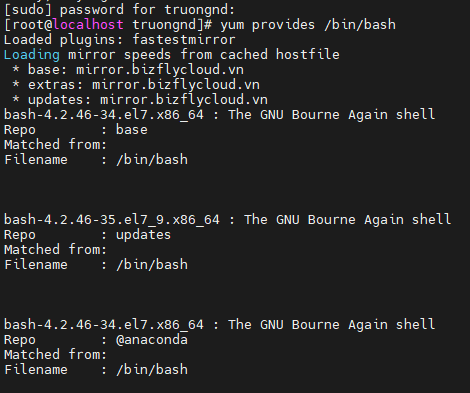


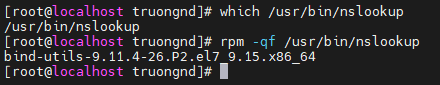


****

**Tìm kiếm package chứa một file ngẫu nhiên**

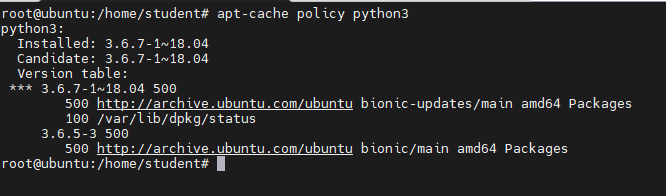
****

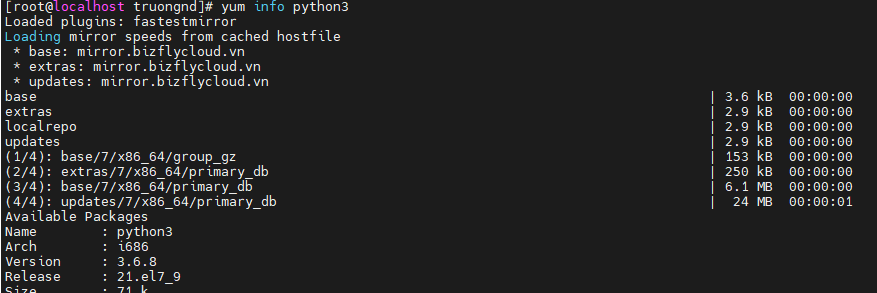
****

****

****

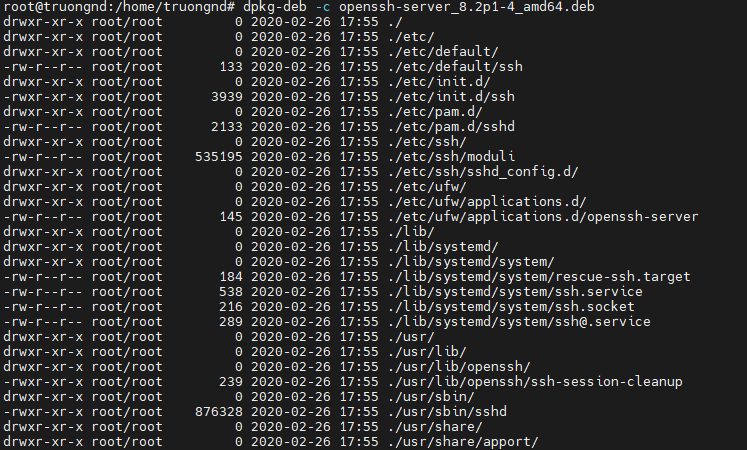
**Tìm kiếm các package đang available trên hệ thống**

****

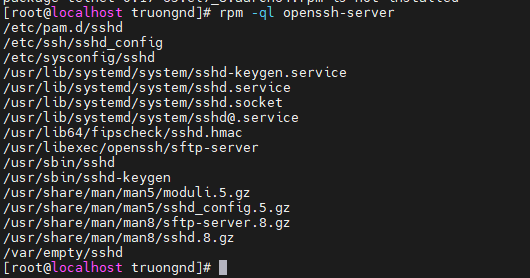
****

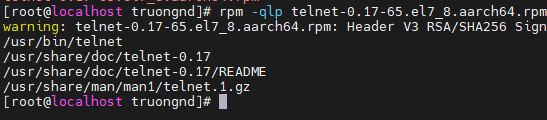
**Liệt kê các file trong 1 package**

**Debian**

****

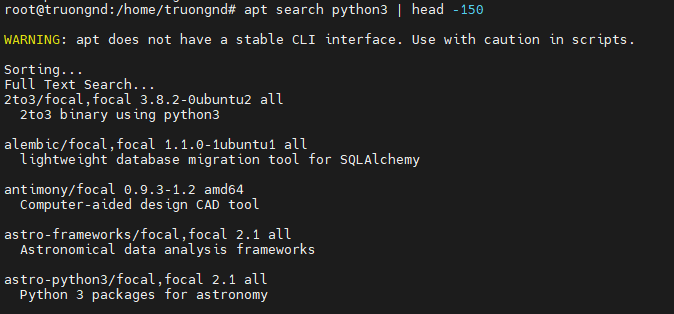
**Redhat**

****

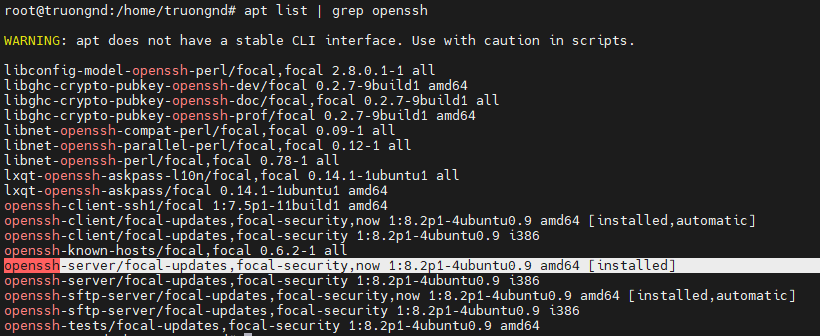
****

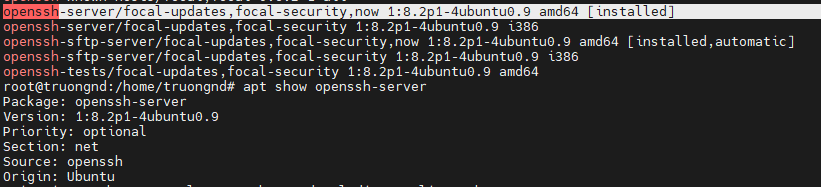
**Tìm kiếm package**

**Debian**



Lệnh trên hiển thị các gói phần mềm liên quan tới Python3 đang có sẵn trong hệ thống.





**Redhat**

*yum search package\_name*

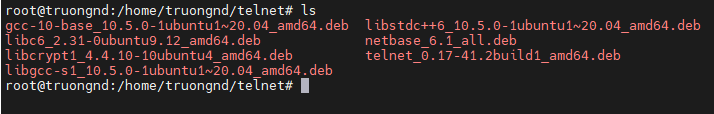
*rpm -qi package\_name*

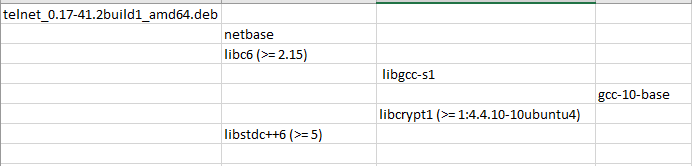
**Install/remove package and denpencies**

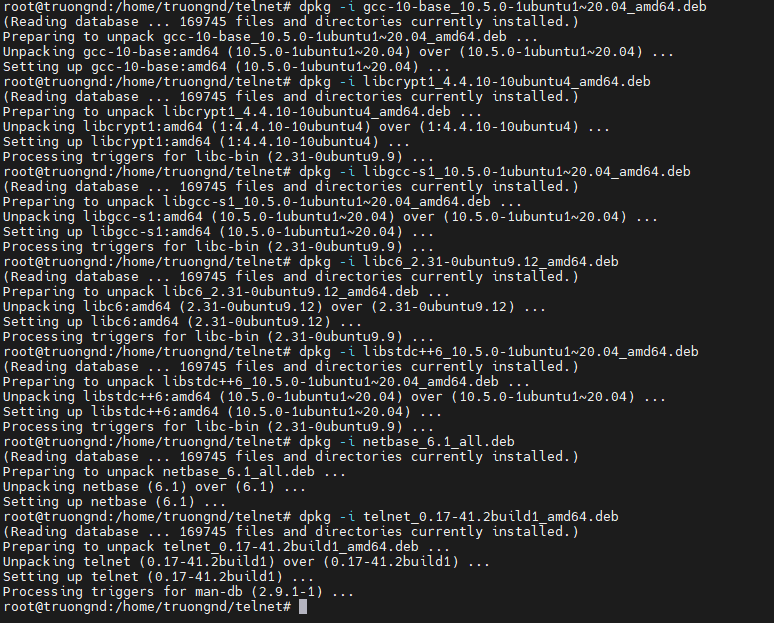
Để intsall một dịch vụ bằng file .deb ta có thể sử dụng công cụ dpkg:

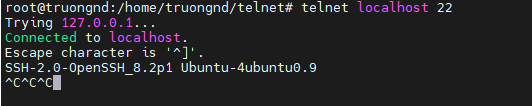


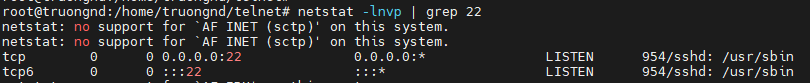
Thực hiện cài đặt các dependency trước khi cài đặt gói telnet.







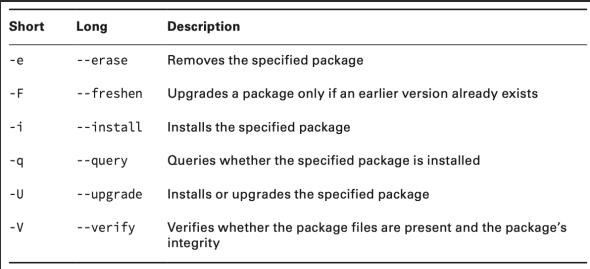
****

****

Kết quả đã thấy được dịch vụ telnet chạy.

Đối với distro redhat cũng tương tự khi ta cần xác định các package dependency sau đó cài đặt chúng sử dụng thông qua công cụ rpm.

***rpm action [option] pkg-file***



**Upgrade packages**

* Để upgrade package có thể sử dụng công cụ apt:

*sudo apt update*

*sudo apt install --only-upgrade package\_name*

* Nếu sử dụng Redhat thì sử dụng công cụ Yum:

*sudo yum update package\_name*

**Liệt kê các package đã được cài đặt**

* Debian:

*dpkg –l*

*apt list –installed*

* Redhat:

*yum list installed*

*rpm -qa*

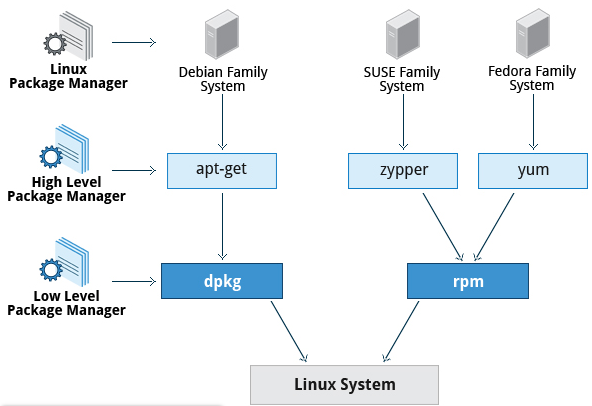
## YUM và APT

Mỗi trình quản lý gói sử dụng một phương pháp khác nhau nhưng cả 2 đều theo dõi những thông tin tương tự:

* Application files: cơ sở dữ liệu gói theo dõi từng tệp riêng lẻ cũng như thư mục chứa nó.
* Library Dependency: Cơ sở dữ liệu gói theo dõi những thư viện được yêu cầu cho mỗi ứng dụng.
* Application version: theo dõi version của ứng dụng.

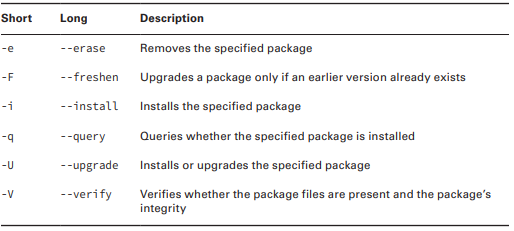
Trong Linux, các công cụ hỗ trợ cài đặt, gỡ bỏ, truy vấn các package trong hệ thống được chia làm 2 level:

* Low level (**dpkg, rpm**): cho phép tương tác trực tiếp với gói cài đặt tuy nhiên sẽ có hạn chế là không thể tự động cài đặt các dependency package (mặc dù có thể liệt kê được các dependency)
* High level (**apt, yum**): khắc phục hạn chế của low level khi tự động tìm kiếm các dependency và cài đặt chúng.



**Using RPM**

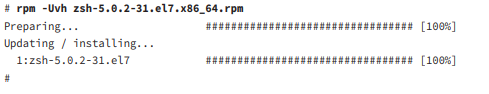
*rpm ACTION [ OPTION ] PACKAGE-FILE*

******

Cài đặt và update RPM package

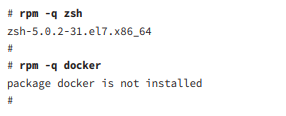
Để cài đặt một package ta sử dụng câu lệnh: *rpm –ivh package\_name*

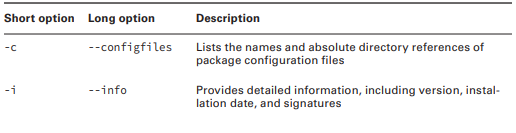
Với option –U dùng để update gói phần mềm:

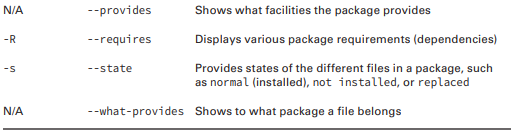


Querying RPM packages

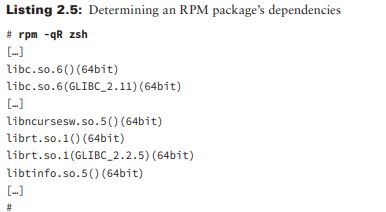
Để xem thông tin về một gói phần mềm ta sử sụng option –q. Với option –q ta có thêm nhiều tùy chọn để xem chi tiết về gói phần mềm.

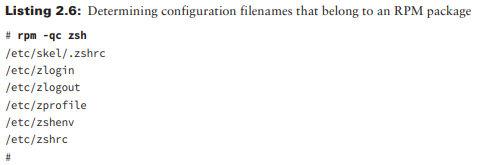


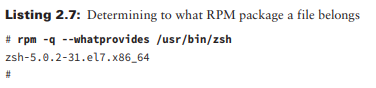








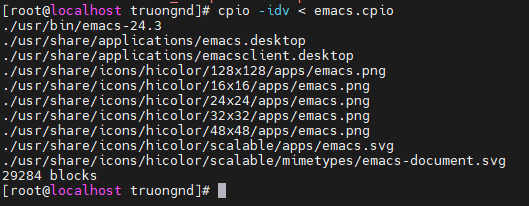




**Extracting Data From RPM**

Đôi khi bạn cần trích xuất các tệp từ package RPM mà không cần cài đặt nó. Tiên ích *rpm2cpio* giúp ta làm việc này.

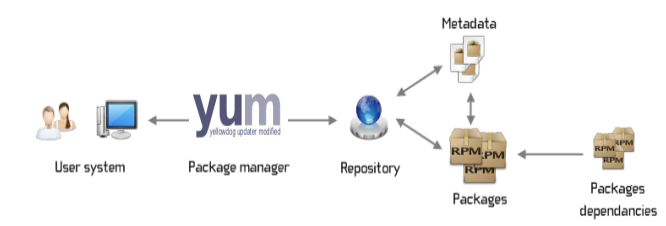






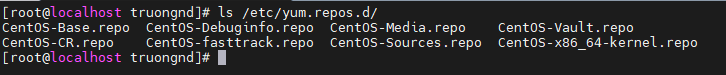
Từ đây ta có thể lấy được file chạy của ứng dụng.

**Using YUM**

****

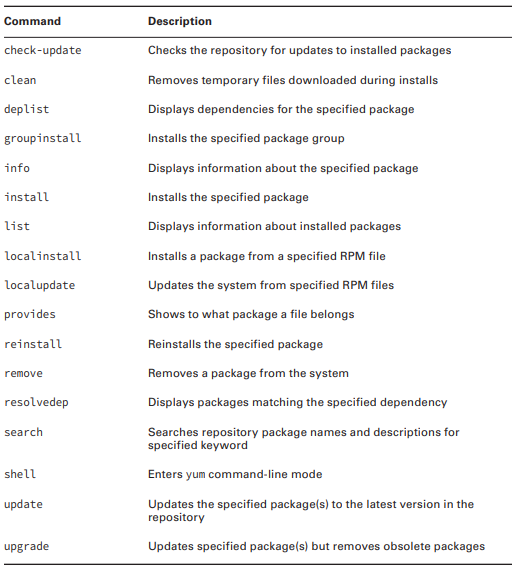
Yum là công cụ high level giúp tự động tìm các dependencies và cài đặt chúng. Lệnh Yum cho phép bạn truy vấn, cài đặt và xóa các gói phần mềm trên hệ thống từ kho lưu trữ (repository).

Lệnh yum sử dụng thư mục */etc/yum.repos.d/* để chứa các repo.



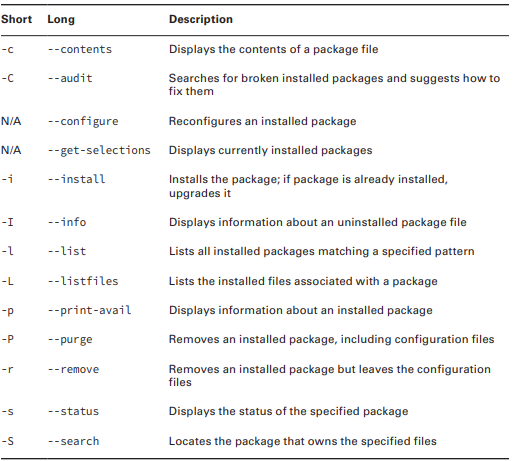
Chúng ta cũng có thể tự thêm một repo vào và sử dụng.

*yum [OPTIONS] [COMMAND] [PACKAGE…]*



**Using dpkg**

*dpkg [option] action pkg-file*

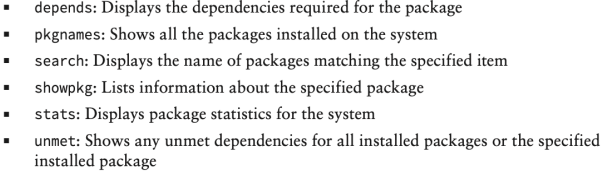
**

**Advanced Package Tool (APT) Suite**

**apt-cache**

Công cụ được sử dụng để truy vấn thông tin về các gói phần mềm:

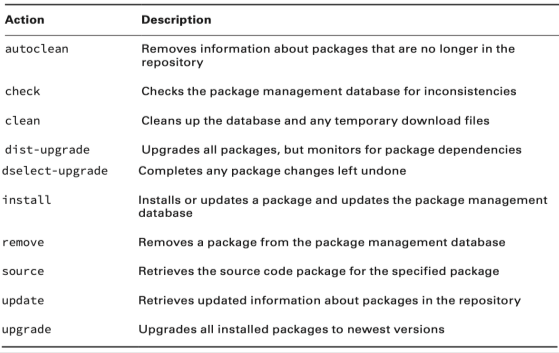
*apt-cache command*

****

|  |  |
| --- | --- |
|  |  |

**apt-get**

*apt-get action*

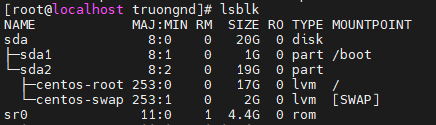
**

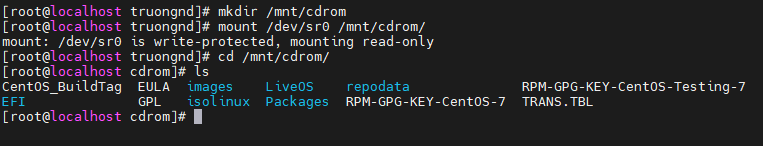
## Cấu hình repository

Trong trường hợp server không kết nối được internet khiến việc cài đặt phần mềm có thể bị lỗi. Chúng ta có thể add một repository và sử dụng. Bạn chỉ cần cấu hình cấu hình để server nhận thư mục là một repo, sau đó có thể cài đặt các gói phần mềm như bình thường.

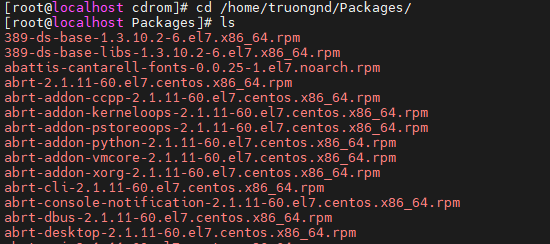
### Cài đặt repo trên CentOS

Bước 1: Mount DVD

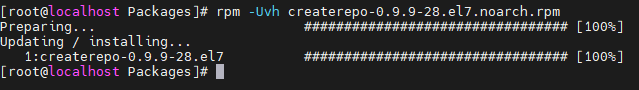




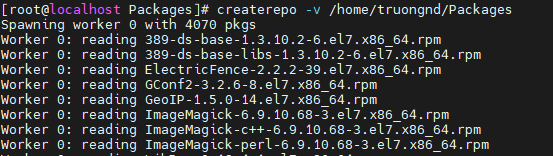




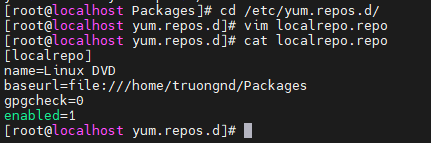
Bước 2: Cài đặt gói createrepo



Bước 3: Tạo repo cho Yum



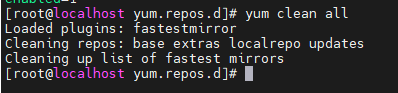
Di chuyển tới thư mục /etc/yum.repos.d/ và tạo file file .repo:



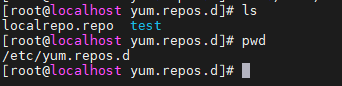
gpgcheck=0: Điều này cho biết xác thực GPG đã được tắt (set bằng 0)

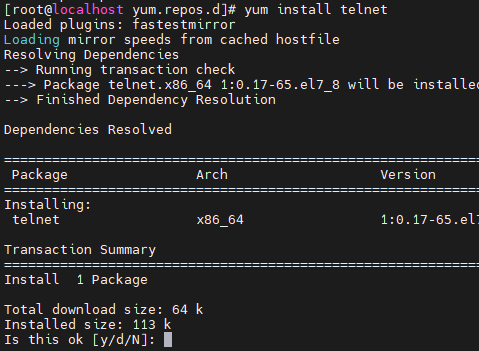
enabled=1: Dòng này cho biết repository đã được kích hoạt

Bước 4: Xác nhận lại Local Repo



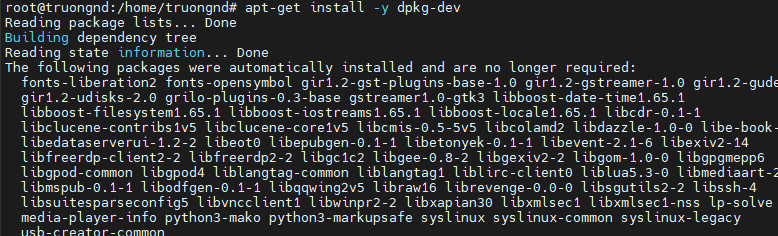
Bước 5: thực hiện test



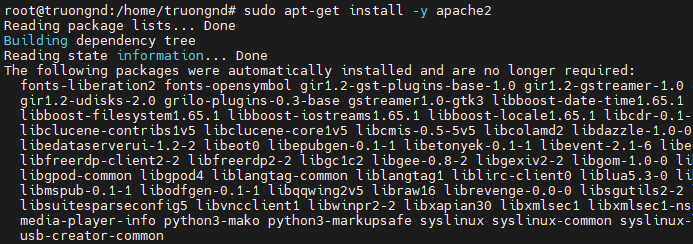


### Cài đặt repo trên ubuntu

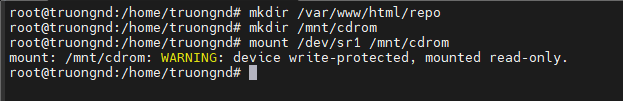
Thực hiện cài đặt gói dpkg-dev để tạo dữ liệu cho repo local

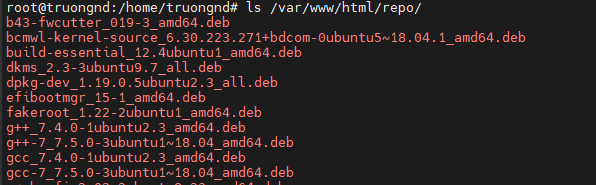


Cài đặt Apache webserver là nơi lưu trữ repo:

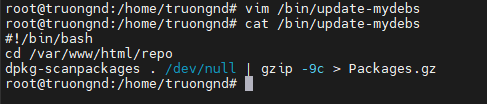


Tìm kiếm tất cả file .deb trong /mnt/cdrom và copy vào thư mục /var/www/html/repo





Viết một script để scan và update repo khi thêm mới các gói .deb vào local repo:

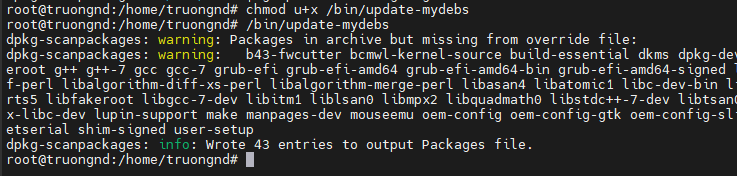


Công cụ *dpkg-scanpackages* được sử dụng để tạo một danh sách các gói phần mềm có sẵn trong một kho lưu trữ. Danh sách này có thể được sử dụng để tạo các gói phần mềm mới hoặc để cập nhật các gói phần mềm hiện có.

Công cụ *dpkg-scanpackages* hoạt động bằng cách quét tất cả các tệp trong kho lưu trữ và tạo một bản ghi cho mỗi gói phần mềm. Bản ghi này bao gồm thông tin sau:

* Tên gói
* Phiên bản gói
* Kiến trúc gói
* Mô tả gói
* Các phụ thuộc của gói

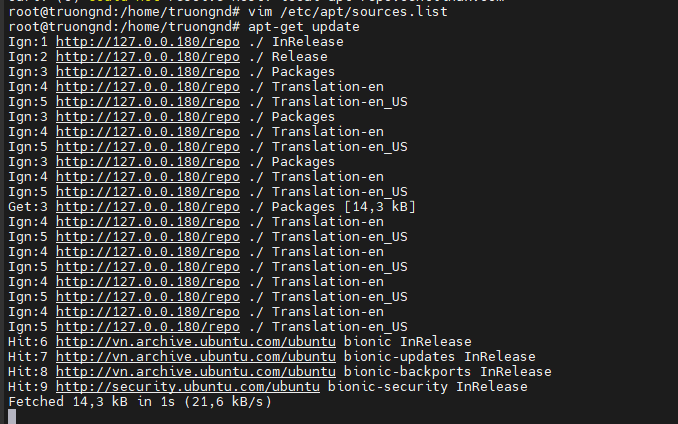
Câu lệnh trên quét các gói cài đặt .deb trong thư mục hiện thời và đầu ra là file /dev/null và cũng lấy output đó nén lại thành file.gz



Add repo vào sources.list







Để xóa một local repo ta thực hiện chỉnh sửa lại file sources.list (thực hiện xóa)

