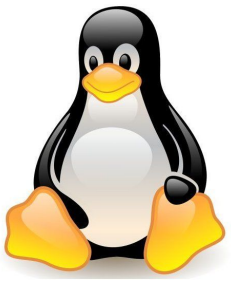
Linux Plus
for
AWS and DevOps



CLARUSWAY
WAY TO REINVENT YOURSELF

Using Package Managers

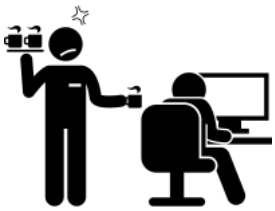
CLARUSWAY
WAY TO REINVENT YOURSELF

Table of Contents

1 Package Management

Package Management

A package manager is a **collection of software tools** that **automates** the process of **installing**, **upgrading**, **configuring**, and **removing** computer programs for a computer's operating system in a consistent manner.



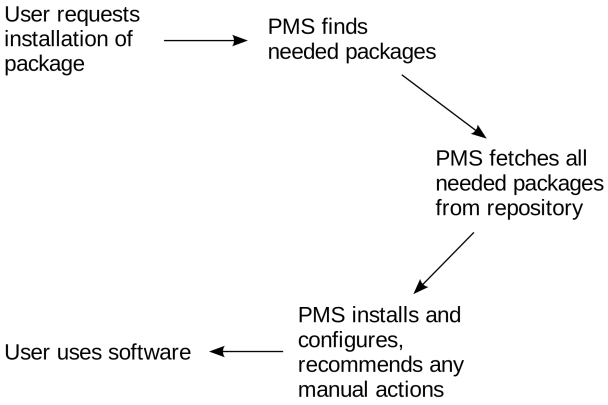
Package Manager → Office Boy

Package Management

A package manager **deals with packages**, **distributions of software** and **data in archive files**. Packages contain **metadata**, such as the **software's name**, **description of its purpose**, **version number**, **vendor**, **checksum**, and a **list of dependencies** necessary for the software to run properly. Upon installation, metadata is stored in a local package database.

Package Management

Package managers are **designed to eliminate the need for manual installs** and **updates**. This can be particularly useful for large enterprises whose operating systems are typically consisting of **hundreds or even tens** of thousands of distinct software packages.



Package Management

Operating System	Format	Tool(s)
Debian	`.deb`	`apt`, `apt-cache`, `apt-get`, `dpkg`
Ubuntu	`.deb`	`apt`, `apt-cache`, `apt-get`, `dpkg`
CentOS	`.rpm`	`yum`
Fedora	`.rpm`	`dnf`
FreeBSD	Ports, `.txz`	`make`, `pkg`

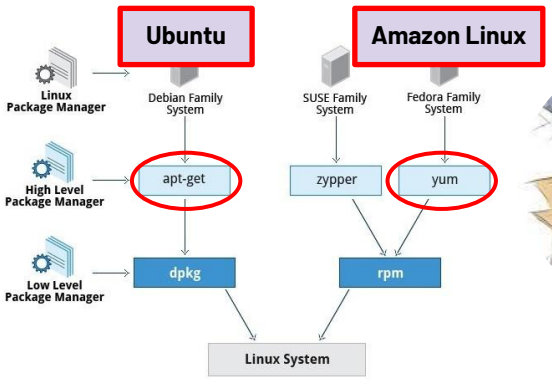
https://stackoverflow.com/questions/10286459/multiple-package-manager

2 Popular Linux System Package Managers

CLARUSWAY
WAY TO REINVENT YOURSELF

Popular Linux System Package Managers

Linux systems use package managers to add or remove the software packages. These **package managers are also a package** so you can install any of them.



CLARUSWAY
WAY TO REINVENT YOURSELF

Popular Linux System Package Managers

Debian Package Managers

dpkg is the **main package management program** for the **Debian Linux** distros. It is used to handle Debian package files with the extension of **.deb**

```
$ dpkg -i [package-name]   # Installing a package
$ dpkg -r [package-name]   #Removing a package
$ dpkg -l                  # Lists installed packages
```


Popular Linux System Package Managers

Debian Package Managers

APT
(Advanced Package Tool)

- The **A**dvanced **P**ackaging **T**ool is what Ubuntu Software Center is built on

- 'apt-get install PACKAGE' will install and organize software
- 'apt-cache list PACKAGE' will search for PACKAGE in the local database
- 'apt-get update' update the local package database

```
$ apt update                  # Update the installed packages
$ apt install [package-name]  # Install a package and all its dependencies
$ apt remove [package-name]   # Remove a package
$ apt purge [package-name]    # Remove a package and its configuration files
```

Popular Linux System Package Managers

Debian Package Managers

Aptitude Package Manager

aptitude tool provides the functionality of **apt-get**, as well as many additional features:

- aptitude provides easy access to all versions of a package
- aptitude tracks of obsolete software
- aptitude has a powerful system for searching particular packages

```
$ aptitude install [package-name]  # Install a package
$ apt-get install [package-name]   # Install a package
```

https://www.tecmint.com/difference-between-apt-and-aptitude/

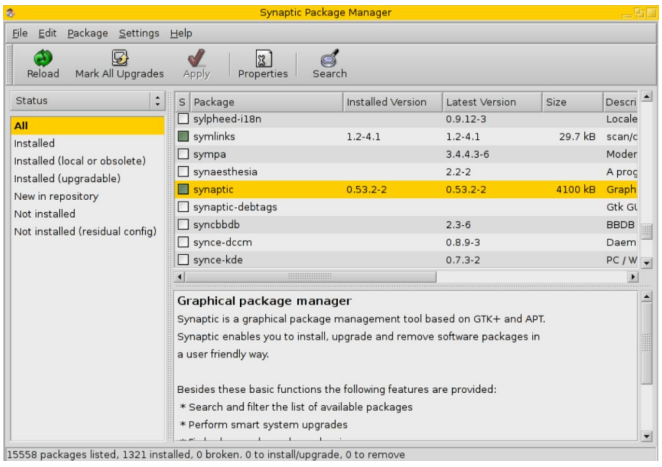
Popular Linux System Package Managers

Debian Package Managers

Synaptic Package Manager

Synaptic is a graphical package manager and used for installing, upgrading and removing single and multiple packages in a more user-friendly way.

```
sudo apt-get install synaptic
```



Popular Linux System Package Managers

Red Hat Package Managers

rpm is the package manager for **Red Hat Linux** operating systems. The installation package files have **.rpm** extension. These files are used for installing programs. **rpm** command has been used for RPM packages by default but new tools are developed for better performance.

```
$ rpm -i [package-name]   # Install a package
$ rpm -e [package-name]   # Uninstall a package
```

Popular Linux System Package Managers

Red Hat Package Managers

YUM (Yellowdog Updater Modified)

YUM is an open-source package manager that was **developed by Duke University**. It is **used** both **in** the **command line and GUI**. It works mostly the same as APT in Debian Linux systems. Here are some examples of YUM.

```
$ yum install [package-name]    # Install a package
$ yum remove [package-name]   # Remove a package
$ yum update [package-name]   # Update a package
```

Popular Linux System Package Managers

Red Hat Package Managers

DNF – Dandified Yum

It is the **new generation of YUM** package manager. It is the default package manager of **Fedora 22 and newer** distros. The usage of DNF is mostly the same as YUM.

```
$ yum install dnf    # Install DNF via yum.
$ dnf –version       # Checking DNF version
$ dnf install        # Installing a package
```


Popular Linux System Package Managers

Red Hat Package Managers

Other RPM tools:

- zypper (openSUSE)
- up2date (Red Hat Enterprise Linux, CentOS 3 and 4, and Oracle Linux)
- urpmi (Mandriva Linux, ROSA Linux, and Mageia
- apt-rpm (Ark Linux,[11] PCLinuxOS and ALT Linux)
- smart (Unity Linux and Fedora)
- rpmquery (Red Hat Enterprise Linux)

Popular Linux System Package Managers

Other Package Managers

Below are a few more notable/interesting package managers.

- Portage: Package manager for Gentoo.
- Pacman: Arch Linux Package manager.
- Nix: A 'Fully Functional/Transactional' package manager.

- **Brew: An Open Source package manager for OSX.**
- **Chocolatey: A package manager for Windows.**

Popular Linux System Package Managers

Other Package Managers

Programming languages have their own default package managers. They help to find and install the packages via searching libraries that exist on the internet for that language.

Examples: **Python: pip** / Ruby: gem, rubygems / Haskell: cabal / NodeJS: npm

PYTHON
PIP

Deep Dive into yum

```
$ yum install [package-name]          # Install a package
$ yum -y install [package-name]       # Skip confirmations during installation
$ yum remove [package-name]           # Remove a package.
$ yum erase [package-name]            # Remove a package (an alias to remove).
$ yum autoremove [package-name] # Remove a package and unused dependencies.
$ yum update [package-name]           # Update a package
$ yum update                          # Update all installed packages
$ yum info [package-name]             # Get information about a package
$ yum list                            # List all installed and available packages
$ yum list [package-name]             # List available matching package(s)
$ yum list installed                  # List installed packages
$ yum --showduplicates list [package-name]  # Lists all available versions
$ yum install [package-name]-[version]          # Install a specific version
```


Exercise 1

Update **all** installed packages

List all installed packages start with **http**

Find all available packages start with **http**

Install **httpd** if available. (Skip confirmations during installation)

List installed **httpd** package

Remove **httpd**

List installed **httpd** package

Exercise 2

Uninstall **git** with all unused dependencies

Check installed **git**

Find previous available **git** version

Install previous available **git** version

Check installed **git** version

Update **git** to the **latest** version

Check installed **git** version

Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

THANKS!

Any questions?