Linux Plus

for

AWS and DevOps

Using Package Managers

Table of Contents

- Installing New Software
- Package Management
- Package Terminology
- Popular Linux System Package Managers

1

Installing New Software

Installing New Software

Most Linux distributions contain **ample support for video and network cards, monitors and other external devices**, so there is usually **no need to install extra drivers**.

If you just can't find what you need, **maybe it is not installed** on your system. Linux moves fast, and software improves on a daily basis.

The **website of your Linux distribution** is a good place to start looking **for additional software** and contains instructions about how to install it on your type of Linux.

https://git-scm.com/downloads

2 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.

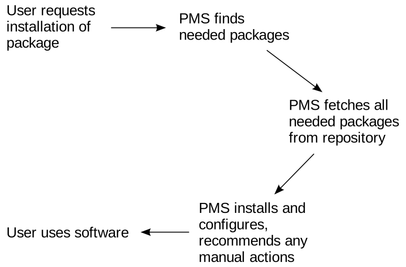
ubuntu
fedora
archlinux
.deb
yum
pacman

Package Management

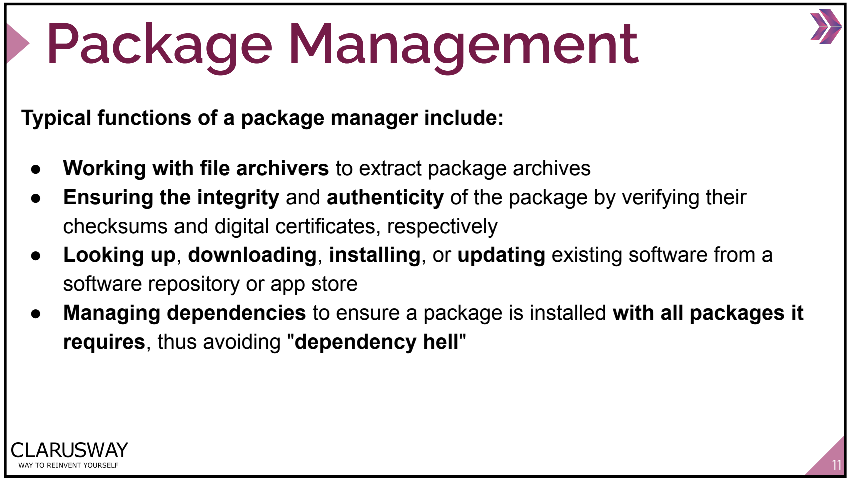
Package managers typically **maintain a database of software dependencies** and **version information** to **prevent** software **mismatches** and **missing prerequisites**. They work closely with software repositories, binary repository managers, and app stores.

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.



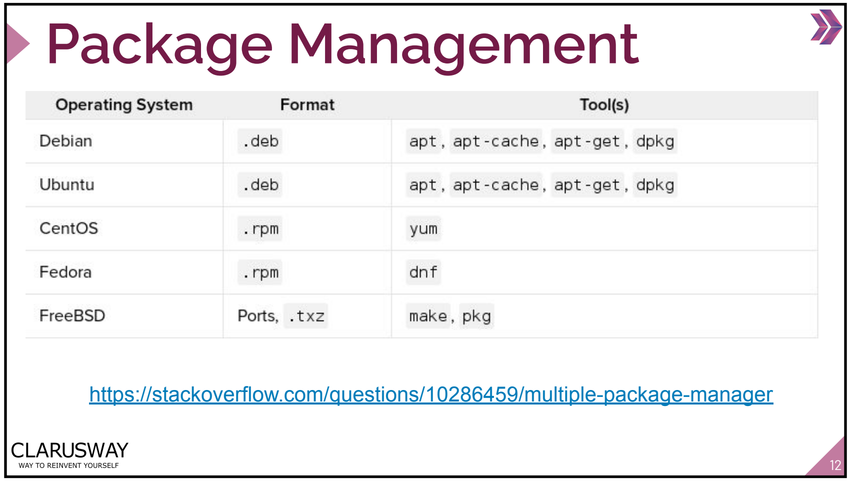
User requests installation of package → PMS finds needed packages → PMS fetches all needed packages from repository → PMS installs and configures, recommends any manual actions → User uses software



Package Management

Typical functions of a package manager include:

- **Working with file archivers** to extract package archives
- **Ensuring the integrity** and **authenticity** of the package by verifying their checksums and digital certificates, respectively
- **Looking up**, **downloading**, **installing**, or **updating** existing software from a software repository or app store
- **Managing dependencies** to ensure a package is installed **with all packages it requires**, thus avoiding "**dependency hell**"

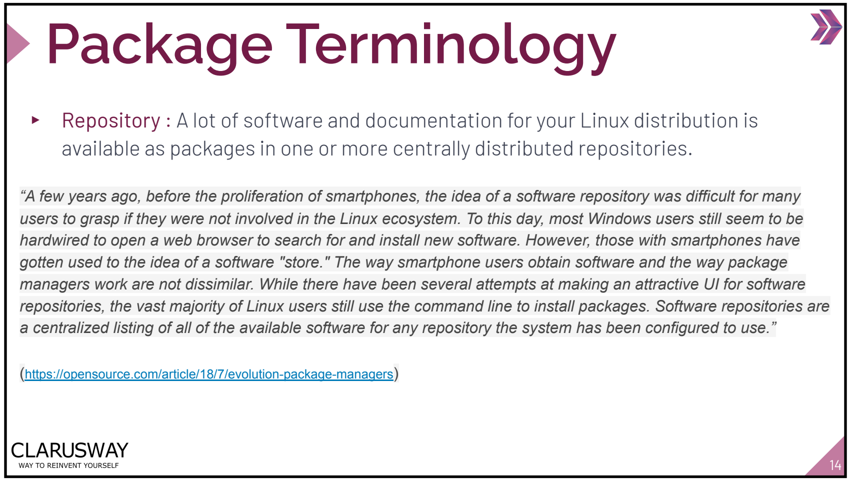


Package Management

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

3 Package Terminology



Package Terminology

- **Repository :** A lot of software and documentation for your Linux distribution is available as packages in one or more centrally distributed repositories.

“A few years ago, before the proliferation of smartphones, the idea of a software repository was difficult for many users to grasp if they were not involved in the Linux ecosystem. To this day, most Windows users still seem to be hardwired to open a web browser to search for and install new software. However, those with smartphones have gotten used to the idea of a software "store." The way smartphone users obtain software and the way package managers work are not dissimilar. While there have been several attempts at making an attractive UI for software repositories, the vast majority of Linux users still use the command line to install packages. Software repositories are a centralized listing of all of the available software for any repository the system has been configured to use.”

(https://opensource.com/article/18/7/evolution-package-managers)

Package Terminology

- Repository

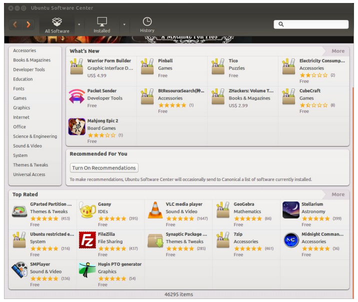
```
user@arch ~ $  aurman -Ss kate

[user@centos ~]$ yum search kate

user@ubuntu ~ $ apt search kate
```

Package Terminology

- .deb Packages : Debian, Ubuntu, Mint and all derivatives from Debian and Ubuntu use .deb packages.
- .rpm Packages : Red Hat, Fedora, **CentOS**, OpenSUSE, Mandriva, Red Flag and others use .rpm packages. The tools to manage software packages on these systems are **yum** and rpm.
- dependency : Some packages need other packages to function. Tools like apt-get, aptitude and yum will install all dependencies.

Package Terminology

- Open Source : These repositories contain a lot of independent open source software. Most distributions also offer this modified source code as a package in one or more source repositories.



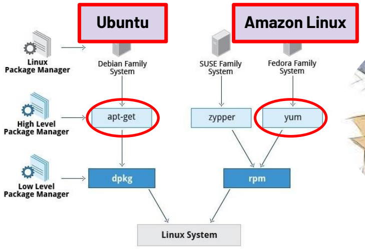
To make recommendations, Ubuntu Software Center will occasionally send to Canonical a list of software currently installed.

4

Popular Linux System Package Managers

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. It is important to understand fully how Linux handles packages.



Ubuntu

Amazon Linux

Linux Package Manager

Debian Family System

SUSE Family System

Fedora Family System

High Level Package Manager

apt-get

zypper

yum

Low Level Package Manager

dpkg

rpm

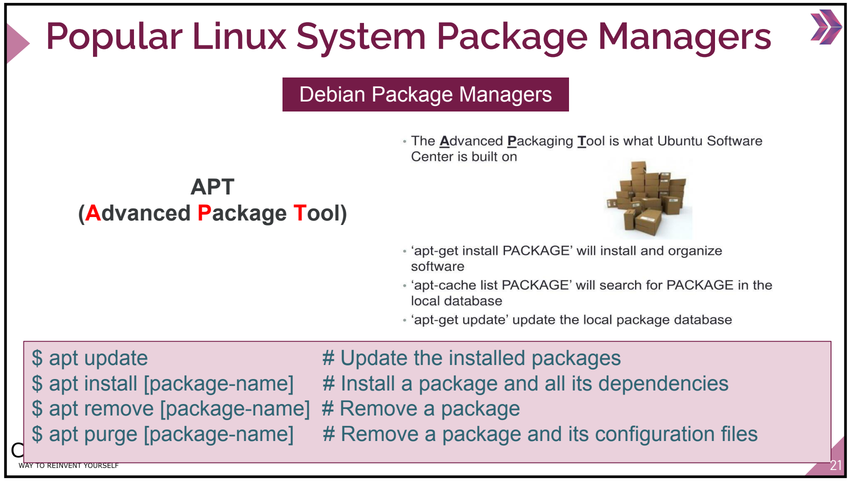
Linux System

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
```



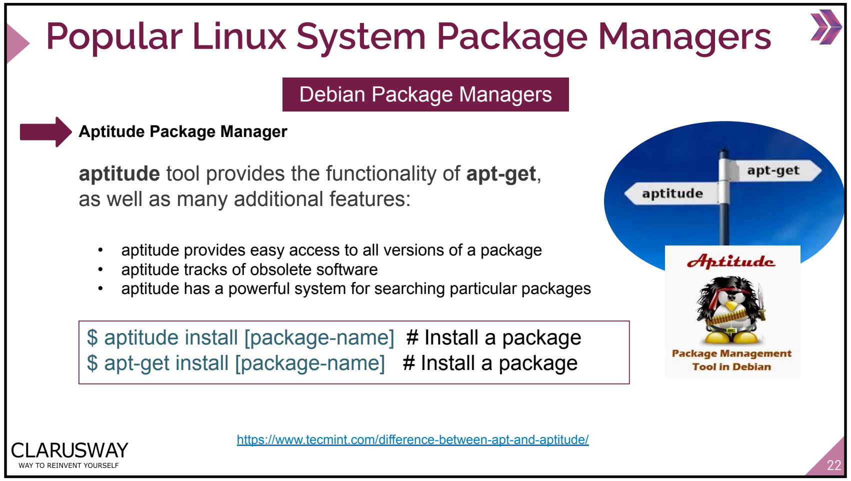
Popular Linux System Package Managers

Debian Package Managers

APT
(Advanced Package Tool)

- The Advanced Packaging Tool 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
```

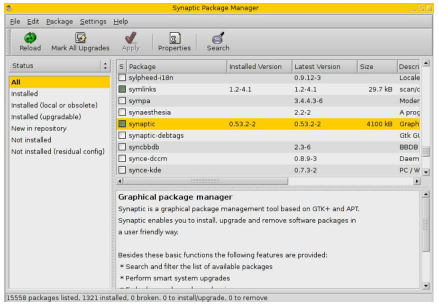
https://www.tecmint.com/difference-between-apt-and-aptitude/

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.



S	Package	Installed Version	Latest Version	Size	Descri
□	sylpheed-i18n		0.9.12-3		Locale
■	symlinks	1.2-4.1	1.2-4.1	29.7 kB	scan/c
□	sympa		3.4.4.3-6		Moder
□	synaesthesia		2.2-2		A proc
■	synaptic	0.53.2-2	0.53.2-2	4100 kB	Graph
□	synaptic-debtags				Gtk GU
□	syncbbdb		2.3-6		BBDB
□	synce-dccm		0.8.9-3		Daem
□	synce-kde		0.7.3-2		PC / W

Graphical package manager

Synaptic is a graphical package management tool based on GTK+ and APT.
Synaptic enables you to install, upgrade and remove software packages in
a user friendly way.

Besides these basic functions the following features are provided:
* Search and filter the list of available packages
* Perform smart system upgrades

15558 packages listed, 1321 installed, 0 broken. 0 to install/upgrade, 0 to remove

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

CLARUSWAY
WAY TO REINVENT YOURSELF

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
```

CLARUSWAY
WAY TO REINVENT YOURSELF

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 supports numerous repositories. 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)

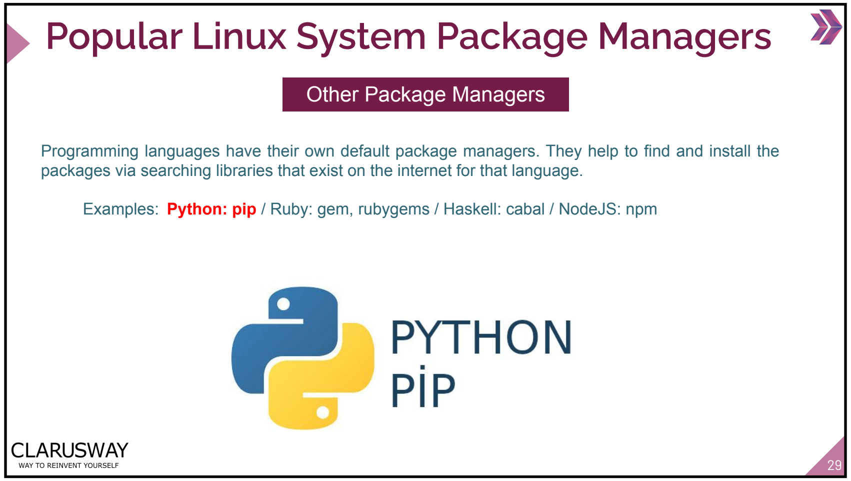
Popular Linux System Package Managers

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.

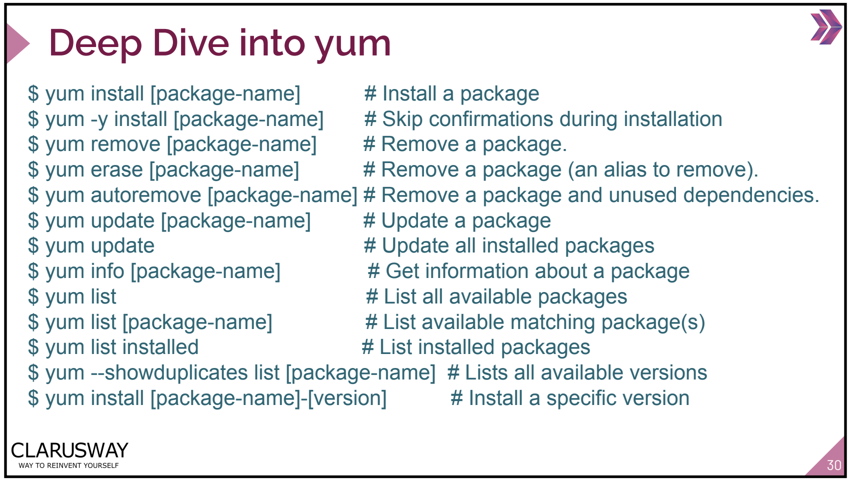


Popular Linux System Package Managers

Other Package Managers

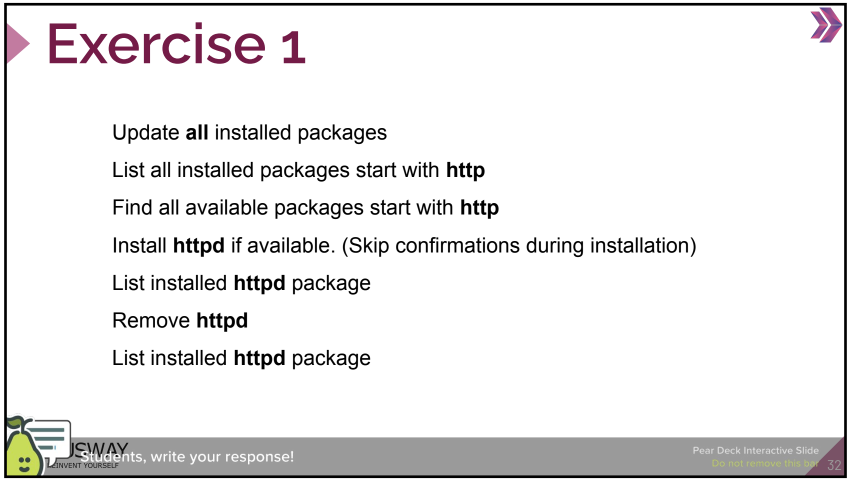
Programming languages have their own default package managers. They help to find and install the packages via searching libraries that exist on the internet for that language.

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



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 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)

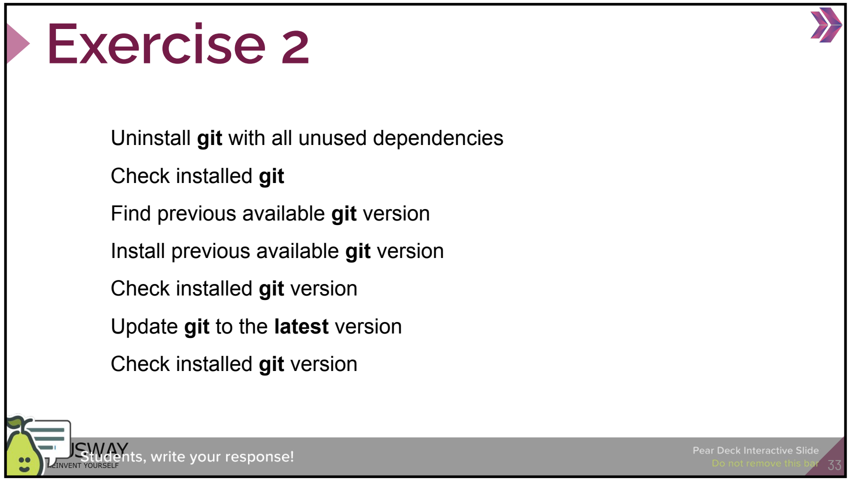
List installed **httpd** package

Remove **httpd**

List installed **httpd** package

Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar



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

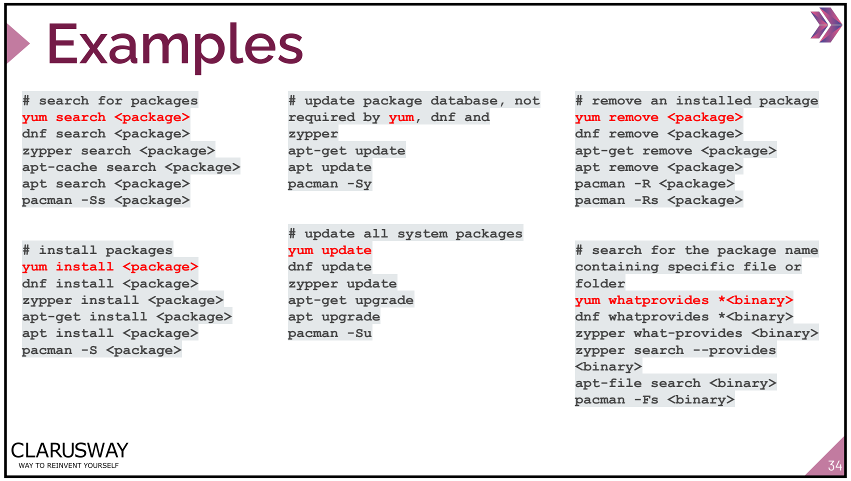
Update **git** to the **latest** version

Check installed **git** version

Students, write your response!

Pear Deck Interactive Slide

Do not remove this bar



Examples

```
# search for packages
yum search <package>
dnf search <package>
zypper search <package>
apt-cache search <package>
apt search <package>
pacman -Ss <package>
```

```
# install packages
yum install <package>
dnf install <package>
zypper install <package>
apt-get install <package>
apt install <package>
pacman -S <package>
```

```
# update package database, not
required by yum, dnf and
zypper
apt-get update
apt update
pacman -Sy
```

```
# update all system packages
yum update
dnf update
zypper update
apt-get upgrade
apt upgrade
pacman -Su
```

```
# remove an installed package
yum remove <package>
dnf remove <package>
apt-get remove <package>
apt remove <package>
pacman -R <package>
pacman -Rs <package>
```

```
# search for the package name
containing specific file or
folder
yum whatprovides *<binary>
dnf whatprovides *<binary>
zypper what-provides <binary>
zypper search --provides
<binary>
apt-file search <binary>
pacman -Fs <binary>
```

THANKS!

Any questions?