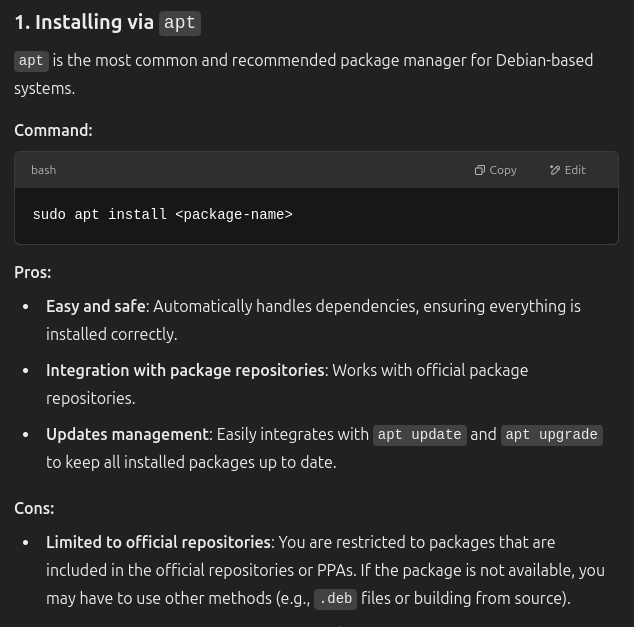
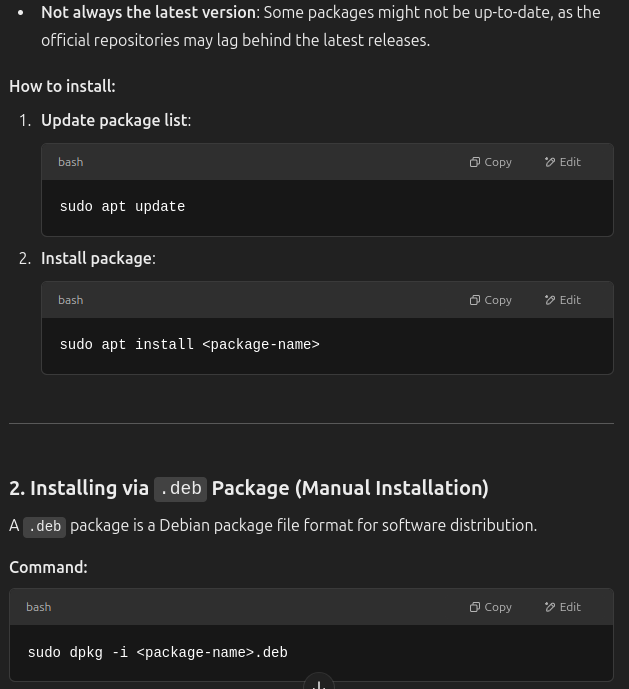
**INSTALL THE RPI-IMAGER**

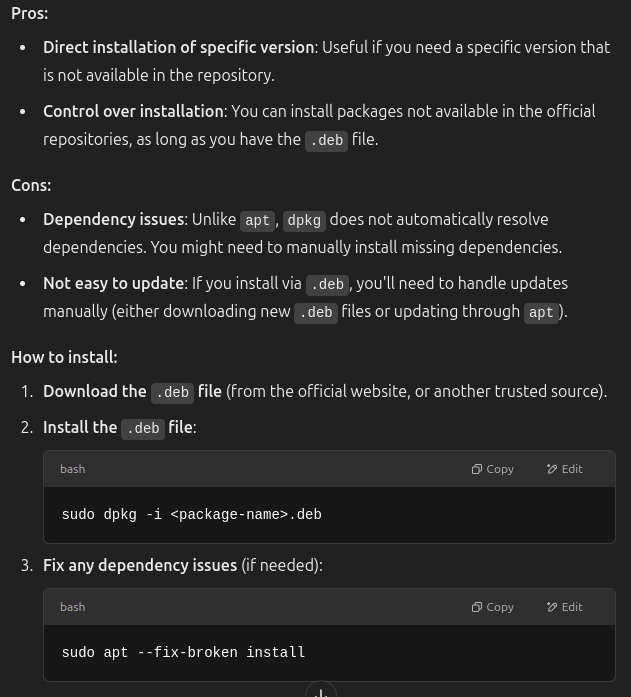
1. Google Chrome , download the “raspberry-pi imager” from the download from the internet. Installing using the commands:



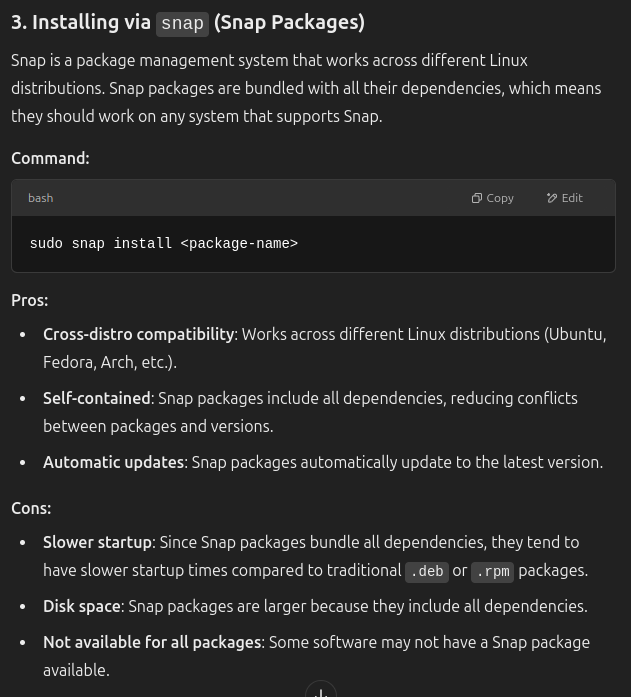




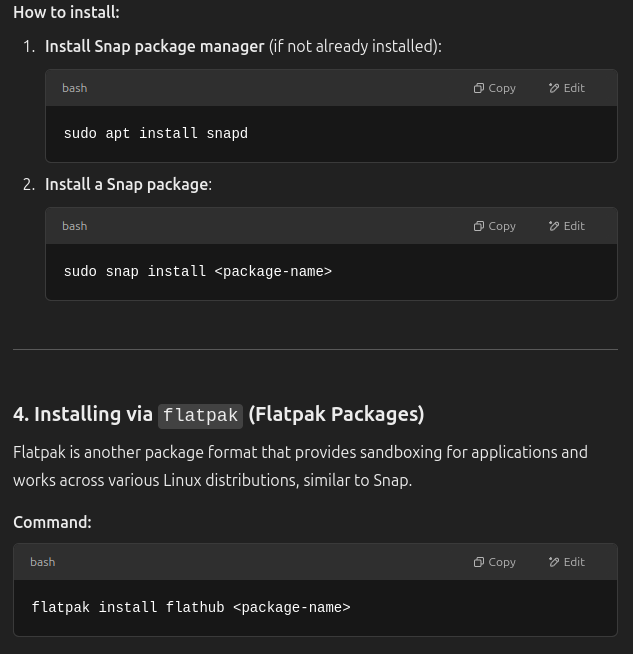








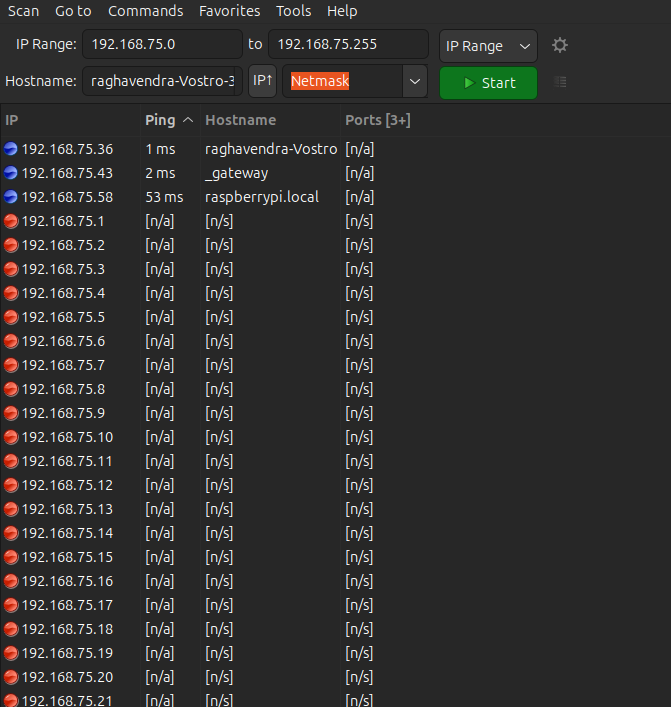




* 1. Download the file and go to the place and then open terminal, then:
  2. sudo apt install name of the package

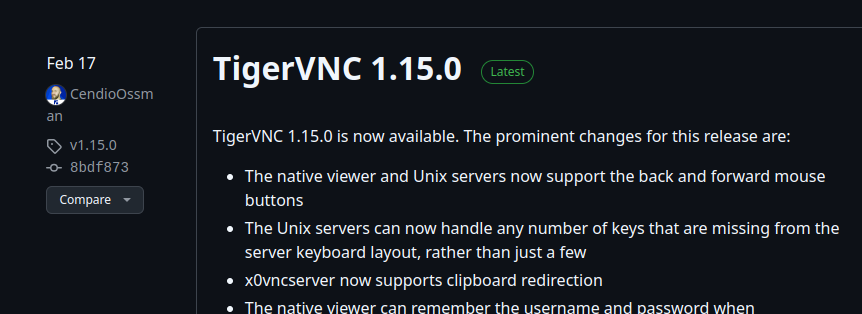
**INSTALL ANGRY – IP SCANNER**

1. Configure what are all the ips that are connected to the same hotspot. So that we can use this toconnect the vcn.



**INSTALL VNC TO THE RASPBERRY PI**

1. In terminal, to be made changes with the raspberry pi:
   1. ssh pi@ip\_address\_found\_in\_previous
   2. Enter password
   3. sudo raspi-config
   4. Go to interface options, then go to vnc, enable the sever
   5. System options
      1. On boot/auto login then Desktop autologin, press on this option.
      2. Then reboot.
   6. Now server is set in the raspberry pi end.
   7. So when rebooting, it will reboot with the vnc server.
   8. In ubuntu side, we need **vnc client**:
      1. Using this we connect to raspberry pi.
      2. Use tiger-vnc.
   9. Tiger-vnc:
      1. In downloads we have the github link



* + 1. Click on the github link, then download the things.
    2. Series of commnds to install vncclient is:

# 1. Extract the package

cd ~/Downloads

tar -xvzf tigervnc-1.15.0.x86\_64.tar.gz

# 2. Copy the files to appropriate directories

sudo cp -r tigervnc-1.15.0.x86\_64/usr/\* /usr/

sudo cp -r tigervnc-1.15.0.x86\_64/etc/\* /etc/

# 3. Make sure binaries are executable

sudo chmod +x /usr/bin/vncviewer

sudo chmod +x /usr/bin/Xvnc

sudo chmod +x /usr/bin/tigervncserver

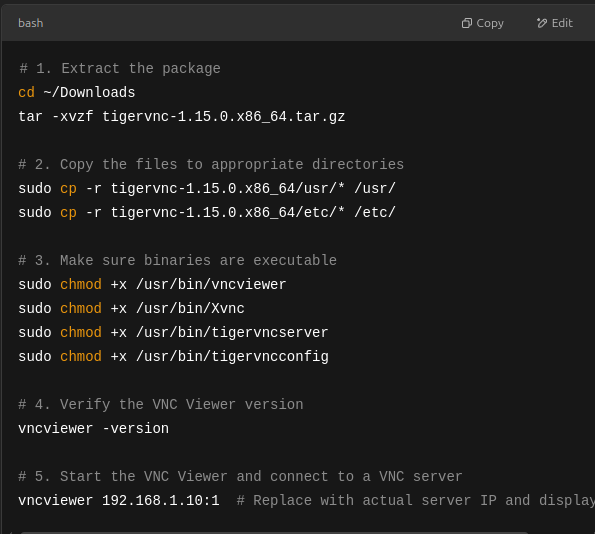
sudo chmod +x /usr/bin/tigervncconfig

# 4. Verify the VNC Viewer version

vncviewer -version

# 5. Start the VNC Viewer and connect to a VNC server

* + - 1. vncviewer 192.168.1.10:1 # Replace with actual server IP and display number



BELOW THINGS ARE NO MORE NEEDED , DO WHAT INSTRUCTOR SAYS.

1. In terminal
   1. sudo su
   2. apt search openssh | grep server
      1. Look for openssh-server
   3. apt install openssh-server
      1. This is used for connecting the virtual machine to the
   4. vncpasswd
      1. Set the 6 digit password.
      2. View only : No.
   5. man vncserver
      1. We do not want everything to run in the background, instead, it should show the output at the front end.
   6. vncserver –fg
      1. This will print the input and output in the front end.
   7. ~/.ssh:
      1. ~: It is nothing but the home folder.
      2. .ssh is one such folder in the home.
2. s

**FAQ’S**

1. Till now I have not installed vnc on my laptop by own , apt search is it going to serach for every packages in the laptop on its own? Or is it going to find the value in the hub ?

Great question! When you run the command:

apt search vnc

### **What Does This Command Do?**

* **It searches the available package repositories**, not the packages already installed on your system. So, it’s going to look for packages in the **Ubuntu repositories** (or any additional repositories you’ve added, like PPAs) **that are configured on your system**.
* The apt tool uses these **repositories** (which are essentially remote servers or mirrors) to search for and fetch metadata about packages. It does **not** search through the packages already installed on your system unless you explicitly ask it to.

### **How It Works:**

1. **Package Repositories**: When you use the apt search command, it searches the lists of packages that are stored in the repositories that are configured in your system’s /etc/apt/sources.list file or in the .list files within the /etc/apt/sources.list.d/ directory.
2. **What It Finds**: It finds packages in the **repositories** and **not just the ones installed** on your laptop. For example, if you're looking for "vnc" packages, it will search in the official Ubuntu repositories or any additional repositories (like PPAs) you may have added.

### **Is It Searching Locally or Remotely?**

* **Remotely**: It queries the list of available packages that are hosted on the configured repository servers (e.g., the Ubuntu repository servers).
* **Locally**: The apt tool uses locally stored metadata (which is downloaded when you run sudo apt update) to perform this search. So it doesn’t directly search through your laptop’s installed packages but looks at the metadata for **available packages**.

### **Example:**

* When you run apt search vnc, it is **not** searching through every package installed on your system. It is searching for all packages that **match "vnc"** in their name or description in the repositories.
* After running this command, you might get output like this (hypothetical example):
* tigervnc-standalone-server/focal 1.11.0+dfsg-1build2 amd64
* vnc4server/focal 4.1.1+xorg4.3.0-0.1ubuntu3 amd64

### **Conclusion:**

* **apt search** looks for packages **in your system's configured repositories**, not in the currently installed packages on your laptop.
* If you want to find **installed packages**, you would use a different command, such as:
* dpkg -l | grep vnc
* This will list the installed packages that match "vnc" on your system.

Let me know if that clears things up! 😄

**Next Question**

1. “Apt search vnc | grep server”\_\_\_\_what is this doing?

The command you provided:

apt search vnc | grep server

### **Breakdown of the command:**

1. **apt search vnc**:

* This part uses the **apt search** command to search for packages related to **"vnc"** in your system's package repositories.
* It will return a list of packages that contain the word "vnc" in their name or description.

1. **| (Pipe Operator)**:

* The pipe (|) takes the output of the **apt search vnc** command (which is a list of package names and descriptions) and passes it as input to the next command, in this case, **grep server**.

1. **grep server**:

* The **grep** command filters the output and **only shows the lines that contain the word "server"**.
* In this case, it's looking for package names or descriptions that contain "server", helping you narrow down to **VNC server-related packages** (as opposed to clients or other tools).

### **What Does This Do?**

* This command lists all the **VNC server-related packages** available in the repositories on your system. It filters out the packages that contain the word "server" in their name or description.
* If you are looking for a **VNC server package** specifically, this will help you identify it.

### **Example Output (hypothetical):**

If you run this command, the output might look something like this:

vnc4server/focal 4.1.1+xorg4.3.0-0.1ubuntu3 amd64

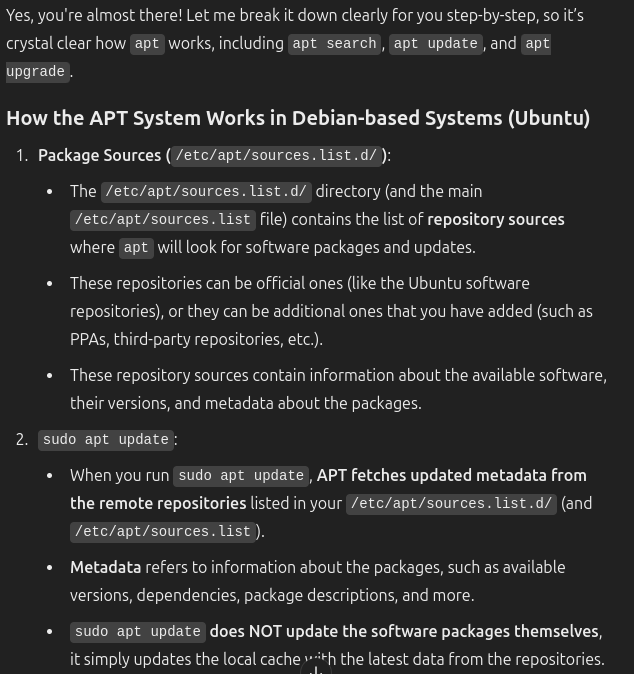
x11vnc/focal 0.9.13-1build1 amd64

tigervnc-standalone-server/focal 1.11.0+dfsg-1build2 amd64

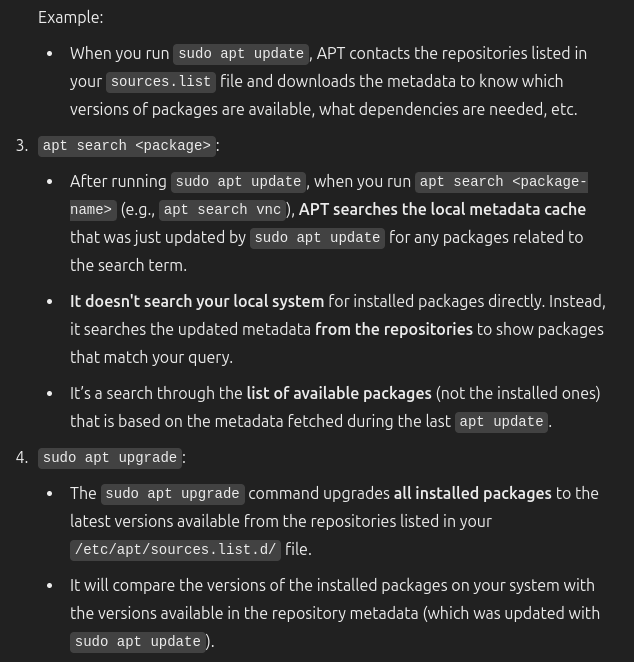
* This shows packages like **vnc4server**, **x11vnc**, and **tigervnc-standalone-server** (all VNC server options).

In short, this command is a useful way to **search for VNC server packages** and narrow down your options quickly!

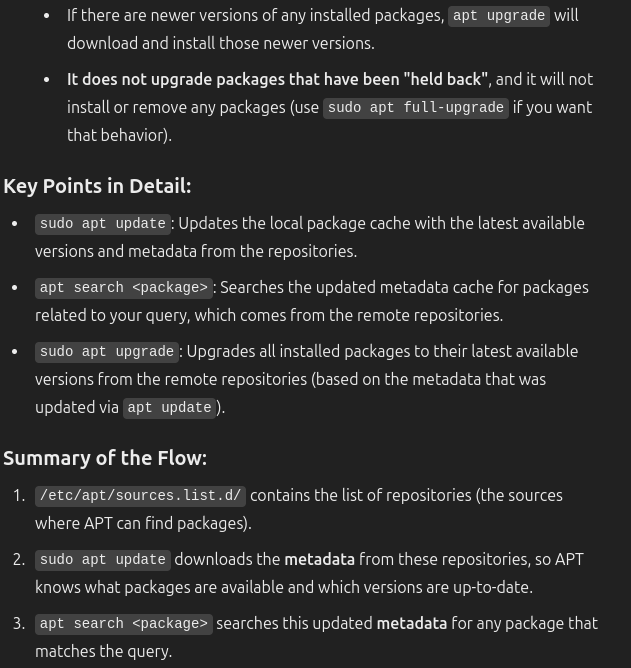
1. How to search for the already installed packages?
   1. dpkg -l | grep vnc — command
   2. dpkg is the Debian package manager. It is used to manage packages on Debian-based systems (such as Ubuntu).
   3. -l is an option that tells dpkg to list all installed packages on your system. It will show you a table of all installed packages, along with their versions, descriptions, and status.
   4. So, dpkg -l will output a list of every package that is installed on your system, whether it's a system package, application, or library.
   5. **| (Pipe Operator)**:
      1. The pipe (|) takes the output of the dpkg -l command (which is a list of installed packages) and passes it as input to the next command (grep).
   6. **grep vnc:**
      1. grep is a command-line utility that searches for a given pattern in text. It will filter through the text and show only the lines that contain the word you’re searching for.
      2. In this case, grep vnc will filter the output from dpkg -l and only show the lines that include the word "vnc". This is useful when you want to check if any VNC-related packages are installed on your system.
2. What are the ways to install?
   1. Method Command Pros Cons
   2. apt sudo apt install <package> Easy, automatic dependencies, updates Limited versions, not all packages
   3. dpkg sudo dpkg -i <package>.deb Install specific versions, more control Dependency issues, no auto updates
   4. snap sudo snap install <package> Cross-distro, bundled dependencies Larger, slower start, limited availability
   5. flatpak flatpak install flathub <package> Cross-distro, sandboxing Larger, slower start, limited availability
   6. Source ./configure; make; sudo make install Latest version, customizable Complex, no updates, manual dependencies
   7. curl/wget `curl -fsSL <url> sudo bash` Simple, fast installation
3. apt search vnc | grep server\_\_\_\_\_so to say , the apt is going to search from the /etc/apt/sources.list.d, so this is updated with the meta data of all the softaware that are necessary for my laptop, when I make sudo apt update, then it will download all the meta data from the remote and list all the necessary required to my laptop to run properly . So when sudo apt upgrade is made, it will upgrade all the installed packages in my laptop. So this will search for everything, that is menioned as of no in the apt folder, right?



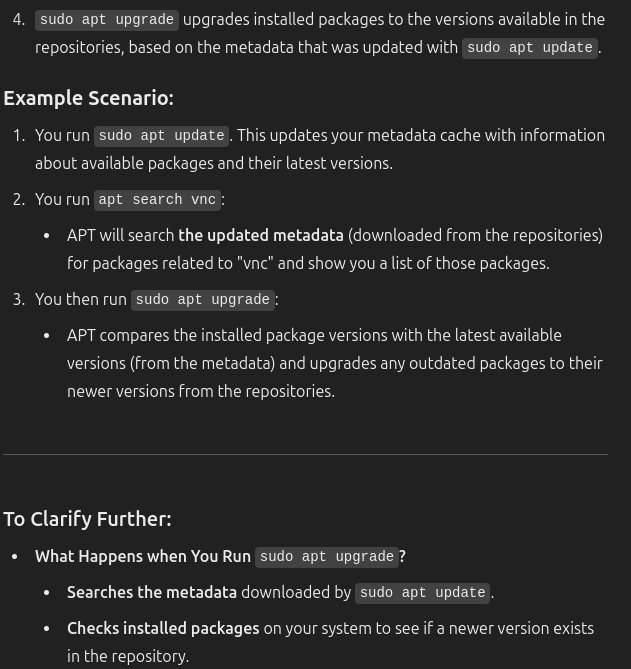




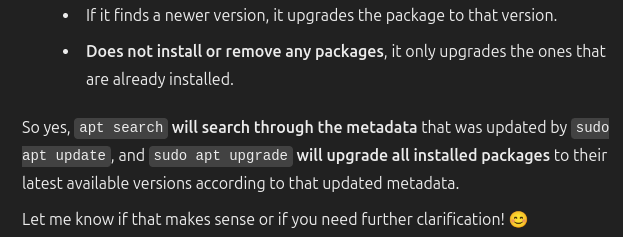












Yes, you're almost there! Let me break it down clearly for you step-by-step, so it’s crystal clear how apt works, including apt search, apt update, and apt upgrade.

### **How the APT System Works in Debian-based Systems (Ubuntu)**

1. **Package Sources (/etc/apt/sources.list.d/)**:

* The /etc/apt/sources.list.d/ directory (and the main /etc/apt/sources.list file) contains the list of **repository sources** where apt will look for software packages and updates.
* These repositories can be official ones (like the Ubuntu software repositories), or they can be additional ones that you have added (such as PPAs, third-party repositories, etc.).
* These repository sources contain information about the available software, their versions, and metadata about the packages.

1. **sudo apt update**:

* When you run sudo apt update, **APT fetches updated metadata from the remote repositories** listed in your /etc/apt/sources.list.d/ (and /etc/apt/sources.list).
* **Metadata** refers to information about the packages, such as available versions, dependencies, package descriptions, and more.
* **sudo apt update does NOT update the software packages themselves**, it simply updates the local cache with the latest data from the repositories.

1. Example:

* When you run sudo apt update, APT contacts the repositories listed in your sources.list file and downloads the metadata to know which versions of packages are available, what dependencies are needed, etc.

1. **apt search <package>**:

* After running sudo apt update, when you run apt search <package-name> (e.g., apt search vnc), **APT searches the local metadata cache** that was just updated by sudo apt update for any packages related to the search term.
* **It doesn't search your local system** for installed packages directly. Instead, it searches the updated metadata **from the repositories** to show packages that match your query.
* It’s a search through the **list of available packages** (not the installed ones) that is based on the metadata fetched during the last apt update.

1. **sudo apt upgrade**:

* The **sudo apt upgrade** command upgrades **all installed packages** to the latest versions available from the repositories listed in your /etc/apt/sources.list.d/ file.
* It will compare the versions of the installed packages on your system with the versions available in the repository metadata (which was updated with sudo apt update).
* If there are newer versions of any installed packages, **apt upgrade** will download and install those newer versions.
* **It does not upgrade packages that have been "held back"**, and it will not install or remove any packages (use sudo apt full-upgrade if you want that behavior).

### **Key Points in Detail:**

* **sudo apt update**: Updates the local package cache with the latest available versions and metadata from the repositories.
* **apt search <package>**: Searches the updated metadata cache for packages related to your query, which comes from the remote repositories.
* **sudo apt upgrade**: Upgrades all installed packages to their latest available versions from the remote repositories (based on the metadata that was updated via apt update).

### **Summary of the Flow:**

1. **/etc/apt/sources.list.d/** contains the list of repositories (the sources where APT can find packages).
2. **sudo apt update** downloads the **metadata** from these repositories, so APT knows what packages are available and which versions are up-to-date.
3. **apt search <package>** searches this updated **metadata** for any package that matches the query.
4. **sudo apt upgrade** upgrades installed packages to the versions available in the repositories, based on the metadata that was updated with sudo apt update.

### **Example Scenario:**

1. You run sudo apt update. This updates your metadata cache with information about available packages and their latest versions.
2. You run apt search vnc:

* APT will search **the updated metadata** (downloaded from the repositories) for packages related to "vnc" and show you a list of those packages.

1. You then run sudo apt upgrade:

* APT compares the installed package versions with the latest available versions (from the metadata) and upgrades any outdated packages to their newer versions from the repositories.

### **To Clarify Further:**

* **What Happens when You Run sudo apt upgrade?**
* **Searches the metadata** downloaded by sudo apt update.
* **Checks installed packages** on your system to see if a newer version exists in the repository.
* If it finds a newer version, it upgrades the package to that version.
* **Does not install or remove any packages**, it only upgrades the ones that are already installed.

So yes, **apt search will search through the metadata** that was updated by sudo apt update, and **sudo apt upgrade will upgrade all installed packages** to their latest available versions according to that updated metadata.

Let me know if that makes sense or if you need further clarification! 😊

1. s