**1. Introduction to Package Management**

**What is Package Management?**

**Package management** is the process of installing, upgrading, configuring, and removing software packages in Linux distributions.

A **package** typically consists of precompiled software, configuration files, and metadata (like dependencies).

**What is APT?**

**APT (Advanced Package Tool)** is a high-level package management system used in **Debian-based distributions** like **Ubuntu**.

APT allows users to interact with software repositories (servers containing software packages) via the command line.

The most common tools used with APT are apt, apt-get, and apt-cache.

**APT vs. dpkg:**

**dpkg** is the lower-level tool that works directly with .deb package files. It is responsible for installing, removing, and providing information about packages, but it does not handle dependencies automatically.

**APT** is a front-end to dpkg and handles dependencies, software repositories, and package downloads.

**2. Installing, Updating, and Removing Software with APT**

**Software Repositories:**

Repositories are collections of software packages stored on a server. **APT** connects to repositories defined in configuration files (usually in /etc/apt/sources.list and /etc/apt/sources.list.d/).

Ubuntu's default repositories include official and community-maintained software packages.

**Updating Package Lists:**

**apt update**: This command updates the local list of available packages and their versions, ensuring that APT knows about the latest packages in the repository.

Example: sudo apt update

APT fetches the latest package lists from the repositories.

**Important**: Run apt update regularly to stay up to date with the latest packages and security patches.

**Upgrading Installed Packages:**

**apt upgrade**: Upgrades all installed packages to their latest available versions, based on the updated package lists.

Example: sudo apt upgrade

It will prompt you to confirm before upgrading packages. You can skip confirmation by using -y to automatically say "yes" to all prompts.

**apt full-upgrade**: A more advanced upgrade command that may remove packages if necessary to upgrade the system completely.

Example: sudo apt full-upgrade

**Use Case**: To handle complex upgrades, such as major version changes or removing obsolete packages.

**Installing Software:**

**apt install**: Installs new packages from the repositories.

Example: sudo apt install firefox

Installs the **Firefox** web browser from Ubuntu's repositories.

If the package is already installed, APT will update it to the latest version.

**Installing Multiple Packages**: You can install multiple packages at once.

Example: sudo apt install firefox vim git

**Search for Software**: To find packages available for installation.

Example: apt search package\_name

Use this command to search for packages available in the repositories.

**Removing Software:**

**apt remove**: Removes a package but leaves its configuration files behind.

Example: sudo apt remove firefox

Removes **Firefox** but keeps configuration files, in case you want to reinstall it later.

**apt purge**: Completely removes a package, including its configuration files.

Example: sudo apt purge firefox

Removes **Firefox** and its configuration files.

**Cleaning Up:**

**apt autoremove**: Removes packages that were automatically installed to satisfy dependencies but are no longer needed.

Example: sudo apt autoremove

**Use Case**: Helps clean up the system by removing unnecessary dependencies.

**apt clean**: Clears the local repository of downloaded package files (.deb files).

Example: sudo apt clean

This is useful for freeing up space, especially if you've installed a lot of software.

**Ping :-**

PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host.

--->This command takes as input the IP address or the URL and sends a data packet to the specified

address with the message “PING” and get a response from the server/host this time is recorded which is called latency.

---->Fast ping low latency means faster connection. Ping uses ICMP(Internet Control Message Protocol) to send an ICMP echo message to the specified host if that host is available then it sends ICMP reply message.

----> Ping is generally measured in millisecond every modern operating system has this ping pre-installed

here **rtt ---> round trip time**

**ping 0**: It is one of the quickest option to ping a localhost.

The terminal will resolve determine the IP address and gives a response once we enter this command.

**ping localhost**: We can use the ping localhost name.

This name will refer to our system and when we enter this command, we will say "ping this system".

**ping -c**----> specifies the number of ICMP echo requests to send to the target host before stopping

the ping process. ex:- ping -c 5 google.com will send 5 ICMP requests and then stop

this time we don't need to mannualy cancel the requests (crtl+c) it automatically

stopped after 4 requests

**ping -s**----->  specifies the size of the ICMP echo request packet in bytes

. For example, ping -s 100 google.com will send an ICMP echo request packet of 100 bytes to google.com. default is 56 bytes

**ping -i**-----> This option sets the time interval between sending packets. By default

, it's set to one second. For example, the following command will send packets every

five seconds

ping -i 5 google.com

**ping -t**------> This option sets the Time-to-Live (TTL) value of the packet, which

determines the number of network hops that the packet can take before being discarded. By default, the TTL value is set to 64. For example, the following command will set the TTL value to 128:   ex:-  ping -t 128 google.com