مازن عبدالتواب سعد

Self-Study

Difference Between APT and DPKG

	DPKG	APT
1)	In Dpkg Packages from external repositories cannot be downloaded.	In APT Packages from external repositories can be downloaded.
2)	DPKG doesn't have such ability.	APT has the ability to resolve dependencies
3)	DPKG also supports local package installation.	APT installs local Packages using dpkg.
4)	Dpkg will terminate and report missing dependencies if you try to install a package with it's dependencies missing.	APT doesn't terminate if you try to install a package whose dependencies are missing. APT downloads the dependencies.
5)	DPKG doesn't support remote package installation.	APT Setup a remote package installation.
6)	The low-level utility dpkg is responsible for installing package contents on the system.	APT is in charge of the system's package listings. It also takes care of package dependencies, ensuring that when one package is installed, all of the others that it requires are likewise installed. It can also get packages from package repositories.
7)	Installs a Debian package file, such as the one you manually downloaded. dpkg -i <package.deb>	Installs or upgrades <package> and all of its dependencies after downloading them. This will also remove a shipment that has been placed on hold. apt-get install <package>

8)	Removes a package called package> that has been installed. dpkg -r <package>	Removes the package <package> as well as any packages that are dependent on it. —purge indicates that packages should be purged; for additional details, see dpkg -P. apt-get remove [--purge] <package>

Difference Between apt and apt-get

	APT	APT-GET
Use cases	apt install <package_name> apt remove <package_name> apt purge <package_name> apt upgrade <package_name> apt update	apt-get install <package_name> apt-get remove <package_name> apt-get purge <package_name> apt-get upgrade <package_name> apt-get update
What is it?	A CLI tool for managing software packages on Debian-based Linux systems.	A CLI tool for managing software packages on Debian-based Linux systems.
Year of release and initial Debian distribution	1998, Debian 2.0 (Hamm) distribution.	2014, Debian 8 (Jessie) distribution.
Search capabilities	No. Users must use apt-cache package management commands instead for search.	Yes.
Dependency resolution	Simple dependency resolution.	Complex dependency resolution with suggested software installations.

Package versions on the file system	Older versions of packages remain on the file system when using apt-get upgrade.	Older versions of packages are deleted from the file system when using apt upgrade.
Printed output from operations	Prints basic output from each apt-get command to the user, without much detail.	Prints verbose output from each apt command to the user for more information on operations, including a progress bar on tasks.

Difference Between RPM and Yum

	RPM	YUM
Origin	Introduced in 1997 by Red Hat.	Upgraded from YUP to YUM in 2003.
Definition	A low-level package manager with basic functionality.	A top-level front-end package manager with advanced functionality.
Dependencies	Does not resolve dependencies.	Resolves and installs package dependencies automatically.
Package installation	Allows multiple package versions to be installed. However, only a single package installation is possible at a time.	Doesn't allow multiple package versions to be installed. Only supports packages available in the repository and shows the already installed packages. On the other hand, YUM can install multiple packages simultaneously.
Upgrades	RPM doesn’t support automatic upgrades.	YUM allows automatic upgrades to the latest available version.
Repository support	RPM doesn't use an online repository for package installation. Instead, it requires the exact local .rpm package	YUM relies on an online repository for installing packages. The utility requires only the package name.

	path to complete the installation.	
Autonomy	RPM is autonomous and utilizes its own database to keep information about the packages on the system.	YUM is a front-end utility that uses the RPM package manager for package management. The utility also uses the RPM database in the backend.
Ease of use	RPM package management and handling gets complicated at times.	It is the easiest way to manage RPM packages.
Rollback	RPM doesn't support change rollback.	YUM allows any changes to be rolled back.