

Install Applications in Debian

tsaimh (2026)
phkoan (2025)

Before We Start: Administrative Privileges

- **root** is the superuser account.
 - It can read, modify, or delete almost anything on the system.
- Avoid staying logged in as root.
 - **A small typo can affect the entire system.**
- Use administrative privileges only when required.
 - Install or remove software
 - Change system-wide settings
 - Manage users, services, disks, and networks

Before We Start: Becoming root

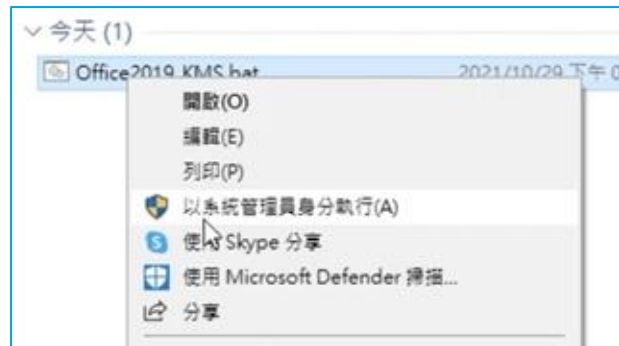
- Check your current identity with `whoami`.
- Use `su -` when you intentionally need a root login shell.
 - Enter the root password.
 - The `-` loads root's login environment.
- On FreeBSD, only user in "wheel" group can use `su` to change to root.
- On Debian, `su` is not restricted to a wheel group by default.
- Prefer `sudo` for routine administration.

```
$ whoami
tsaimh
$ su -
Password:
# whoami
root
```



Before We Start: Why sudo?

- `sudo` runs one command with another user's privileges (root by default).
- Why use it?
 - Least privilege: elevate only when needed
 - Use your own account; administrative actions can be audited
 - Reduce accidental changes from a persistent root shell
- Example: `sudo apt install vim`



Install Sudo

● Debian installer behavior

- Root password left blank → sudo is installed; the first user gets sudo access.
- Root password set → sudo is not installed; the first user is not added to the sudo group.



● Our lab sets a root password, so install sudo manually:

- `su -`
- `apt update`
- `apt install sudo`

● FreeBSD: `pkg install sudo`



Enable Sudo

- Recommended on Debian: grant access through the sudo group.
 - Become root: `su -`
 - Add the user: `usermod -aG sudo <username>`
 - Log out and log back in so the new group membership takes effect.
- Verify the configuration
 - Check groups: `groups`
 - Check sudo rights: `sudo -l`
- For custom policy, use `visudo` — do not edit `/etc/sudoers` directly.

```
# usermod -aG sudo tsaimh
$ sudo -l
```

Use Sudo

- Prefix only the command that needs administrative privileges.
 - `sudo whoami`
 - `sudo apt install vim`
- Use `sudo -l` to see what you are allowed to run.
- Credentials may be cached briefly; the timeout is configurable.
- **Stop and read the command before pressing Enter.**

```
$ whoami
tsaimh
$ sudo whoami
[sudo] password for
tsaimh:
root
$ sudo -l
...
```

Install Software — Overview

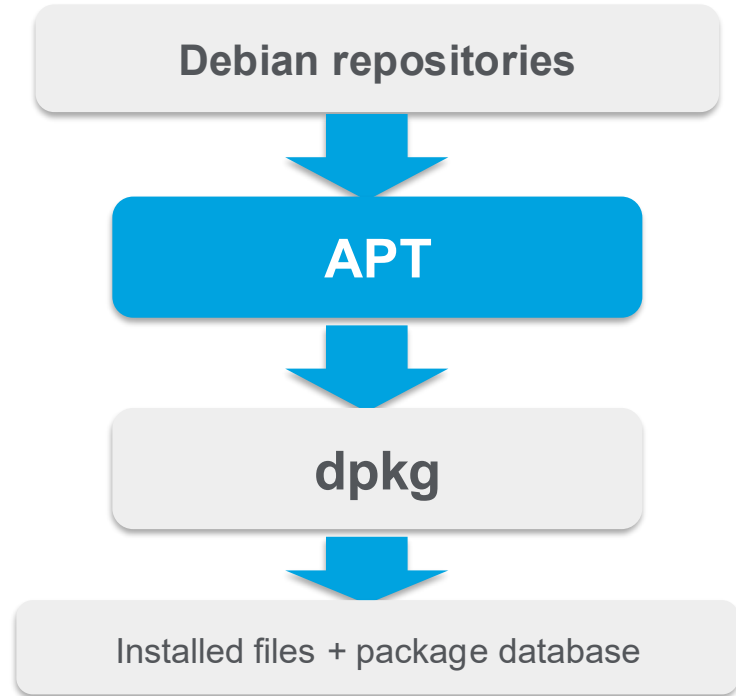
- Package format and package manager are different concepts.
 - Debian / Ubuntu: **.deb** packages; **APT** + **dpkg**
 - RHEL / Fedora: **.rpm** packages; **DNF** + **rpm**
 - Arch Linux: **pacman**
 - FreeBSD: **pkg**
- On Debian, prefer these methods in order:
 - Official Debian repository
 - Trusted vendor repository
 - Local **.deb** file
 - Build from source only when necessary

Install Software from Source

- Use APT when a trustworthy package is available.
- Building from source or installing from a Git repository may be necessary, but you become responsible for dependencies and updates.
- Be cautious with `curl <url> | sh` or similar commands.
 - Remote code is executed immediately with little opportunity to inspect it.
 - A compromised site, script, or dependency can compromise your system.
- Safer workflow: download → inspect → verify checksum/signature → execute.

Package System in Debian

- **APT** is the high-level package interface.
 - Finds packages in repositories
 - Resolves dependencies
 - Downloads, installs, and upgrades packages
- **dpkg** is the low-level Debian package tool.
 - Installs and queries local **.deb** packages
 - Does not resolve repository dependencies by itself



Refresh & Search Packages

- Refresh package metadata before installing or upgrading:
 - `sudo apt update`
 - `apt update` refreshes the package index; it does not upgrade installed software.
- Search and inspect packages:
 - `apt search <keyword>`
 - `apt show <package>`
- Repository definitions are in `/etc/apt/sources.list` and files under `/etc/apt/sources.list.d/`

Package Installation

- Install from configured repositories:
 - `sudo apt install tmux`
- Install a local `.deb` file:
 - `sudo apt install ./package.deb`
 - The `./` (or another path) tells APT that this is a local file.
- Prefer APT over `dpkg -i` for routine installation because APT can resolve dependencies.

```
$ sudo apt install tmux      # repository
$ sudo apt install ./app.deb # local file
```

Third-Party Repositories

- Some vendors publish their own APT repositories (for example, Docker).
- Keep third-party sources in separate files under `/etc/apt/sources.list.d/`
- Use the vendor's signing key with `Signed-By` so trust is scoped to that repository.
- **Only add repositories you trust.**
 - A third-party repository can provide or replace software on your system.
- Follow the vendor's current official installation instructions.
 - Docker repository setup will be covered later in the semester.

Package Upgrade

- Typical update of a Debian stable system:
 - `sudo apt update`
 - `sudo apt upgrade`
- If dependency changes may require adding or removing packages:
 - `sudo apt full-upgrade`
- Upgrade one installed package only:
 - `sudo apt install --only-upgrade <package>`
- For a major Debian release upgrade, follow the official release notes — do not treat `apt full-upgrade` alone as the whole procedure.

Package Removal

- Remove a package but keep modified configuration files:
 - `sudo apt remove <package>`
- Remove a package and its packaged configuration files:
 - `sudo apt purge <package>`
- Reinstall a package:
 - `sudo apt reinstall <package>`
- Remove no-longer-needed automatic dependencies:
 - `sudo apt autoremove`
 - **Always review the proposed removal list before confirming.**

Package Information

- Show package metadata: `apt show <package>`
- Show installed/candidate version and source priority: `apt policy <package>`
- List files installed by a package: `dpkg -L <package>`
- List installed packages: `apt list --installed`

```
$ apt show tmux  
$ apt policy tmux  
$ dpkg -L tmux  
$ apt list --installed
```