📦 Linux Package Management & System Maintenance

📘 Scenario Overview

As the newly appointed **Software Steward**, I was tasked with stabilizing a cluttered development server that had been unmanaged for some time. The objective was to bring the system up to date, install requested tools, verify installations, remove obsolete software, and clean up unused dependencies.

This lab simulates **real-world system maintenance on Debian-based Linux servers**, where clean package management directly impacts security, stability, and performance.

🎯 Objectives

- Update system package repositories
- Install required software for developers
- Verify installed package details and versions
- Remove obsolete software
- Clean up unused dependencies to maintain a lean system

🔄 Step 1: Update System Package Repositories

Purpose

Ensure the system has the latest package metadata to avoid installation errors and security issues.

Command Used

```
sudo apt update
```

Result

- Local package index synchronized with repositories
- System ready for safe and reliable installations

```
labex:project/ $ sudo apt update
Hit:1 http://mirrors.cloud.aliyuncs.com/ubuntu jammy InRelease
Hit:2 http://mirrors.cloud.aliyuncs.com/ubuntu jammy-updates InRelease
Hit:3 http://mirrors.cloud.aliyuncs.com/ubuntu jammy-backports InRelease
Hit:4 http://mirrors.cloud.aliyuncs.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
315 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

📥 Step 2: Install Essential Software (`neofetch`)

Purpose

Provide developers with a tool to quickly view system information.

Command Used

```
sudo apt install neofetch
```

Outcome

- `neofetch` successfully installed
- Package manager resolved and installed dependencies automatically

```
labex:project/ $ sudo apt install neofetch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-libxfce4util-1.0 gir1.2-xfconf-0 gsfonts-x11 libdbus-glib-1-2
  libgdk-pixbuf-xlib-2.0-0 libjpeg-turbo-progs miscfiles xscreensaver-data
```

🔍 Step 3: Verify Installed Package Details

Purpose

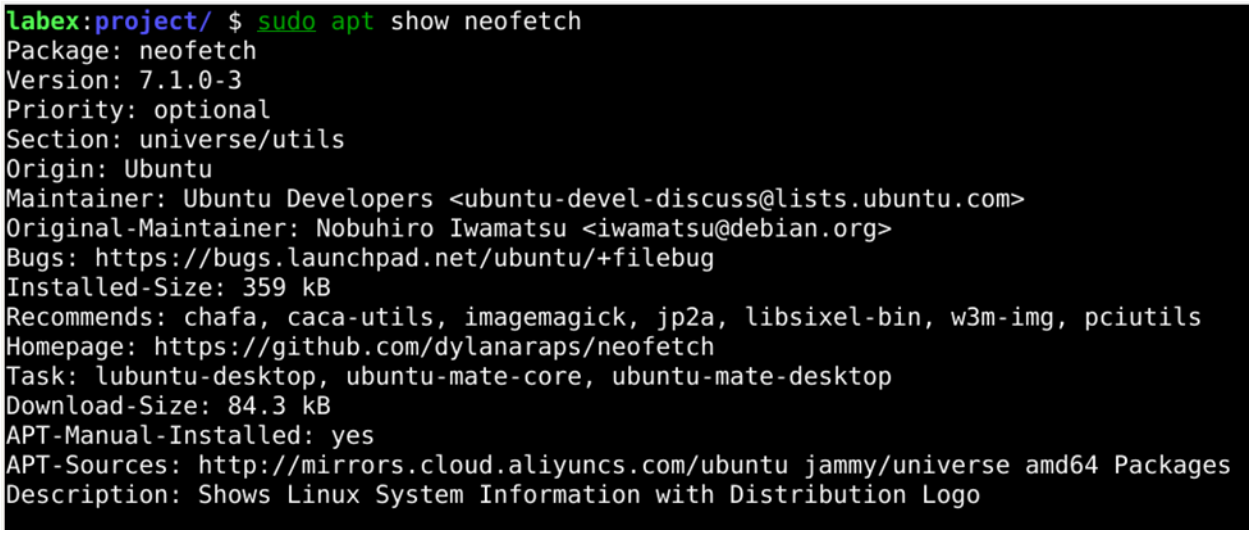
Confirm installation success and document the installed version.

Command Used

```
apt show neofetch
```

Result

- Displayed package metadata
- Verified version number, description, and dependencies



🗑️ Step 4: Remove Obsolete Software (`figlet`)

Purpose

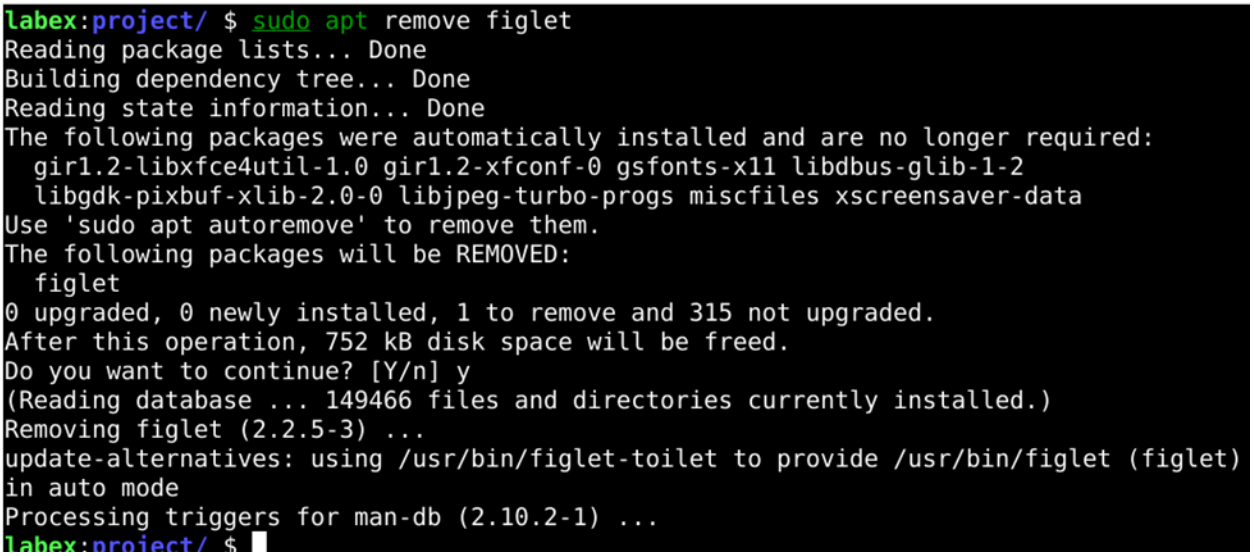
Eliminate unused software to reduce attack surface and clutter.

Command Used

```
sudo apt remove figlet
```

Outcome

- `figlet` package removed
- System no longer carries unnecessary software



🧹 Step 5: Clean Up Unused Dependencies

Purpose

Remove orphaned packages that were installed automatically but are no longer needed.

Command Used

```
sudo apt autoremove
```

Result

- Freed disk space
- Improved system cleanliness and maintainability

```
labex:project/ $ sudo apt autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  gir1.2-libxfce4util-1.0 gir1.2-xfconf-0 gsfonts-x11 libdbus-glib-1-2
  libgdk-pixbuf-xlib-2.0-0 libjpeg-turbo-progs miscfiles xscreensaver-data
0 upgraded, 0 newly installed, 8 to remove and 315 not upgraded.
After this operation, 5,641 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 149393 files and directories currently installed.)
Removing gir1.2-libxfce4util-1.0:amd64 (4.16.0-1) ...
Removing gir1.2-xfconf-0 (4.16.0-2) ...
Removing gsfonts-x11 (0.28) ...
```

🧠 Skills Demonstrated

- Debian-based package management (`apt`)
- Secure system updating practices
- Software installation and verification
- Safe removal of obsolete packages
- Dependency management and cleanup
- Server hygiene and lifecycle maintenance