# 100+ Tips & Tricks — Ubuntu Server 26.04 LTS

**“Resolute Raccoon”**

*Practical tips, commands, and things worth knowing for running, securing, and administering Ubuntu Server 26.04 LTS — organized by area.*

**101 tips · 13 categories · supported until 2031**

*Compiled July 2026 · Released April 23, 2026 · © SplitZrate 2026*

---

## Contents

1. Installation & First Boot . . . . . 2
2. Security . . . . . 2
3. Networking . . . . . 4
4. Package Management & Updates . . . . . 5
5. systemd, Services & Processes . . . . . 6
6. Storage & File Systems . . . . . 6
7. Containers & Virtualization . . . . . 7
8. Cloud & Provisioning . . . . . 8
9. Performance & Monitoring . . . . . 9
10. Backup & Recovery . . . . . 9
11. Remote Access & Administration . . . . . 10
12. Databases & Web Services . . . . . 11
13. Ubuntu Pro, WSL & More . . . . . 11

# 1. Installation & First Boot

### 1.1 Put the new Subiquity installer to smart use

In 26.04 the user account is created earlier in the install flow, which fixes earlier crashes in virtual machines. Let the installer detect and install OEM/HWE drivers automatically when needed — tick the option when offered instead of installing drivers manually afterward.

### 1.2 Automate installs with autoinstall

For repeat server installs, build an `autoinstall.yaml` and serve it via a simple HTTP server or a USB stick. This lets you install identical servers with no interaction. Example: `sudo cloud-init status --wait` to verify the autoinstall process has finished before continuing provisioning.

### 1.3 Choose the minimal install for production servers

At install time, pick the minimal variant if your cloud provider offers it — fewer preinstalled packages means a smaller attack surface and a faster boot.

### 1.4 Check architecture requirements before upgrading

26.04 requires AMD64v3 for cloud images (Intel Ivy Bridge/Sandy Bridge are no longer supported), RVA23S64 for RISC-V, and at minimum IBM Z generation z15 for s390x. Check your hardware before planning an upgrade on older platforms.

### 1.5 Verify cgroup v2 before upgrading from older Ubuntu

systemd 259 in 26.04 has dropped cgroup v1 support entirely. Run `stat -fc %T /sys/fs/cgroup/` before upgrading — if the result isn't `cgroup2fs`, migrate any workloads still relying on cgroup v1 before you upgrade.

### 1.6 Be aware that /tmp is now tmpfs

The `/tmp` directory is now RAM-backed (tmpfs) by default and limited by available memory. Point heavy temporary files (large build jobs, for instance) at a disk-backed directory instead to avoid filling RAM.

### 1.7 Check the new location for removable media

Mount points for removable media have moved from `/media` to `/run/media`. Update any scripts or automation that hardcode the old path.

---

# 2. Security

### 2.1 Know that TPM-backed disk encryption is Desktop-only for now

Ubuntu 26.04's new TPM-bound full-disk encryption (the flow where the key is sealed to the TPM chip and the disk unlocks automatically at boot with no password) shipped for **Desktop**, not Server — don't plan around it for a headless install. If you still want disk encryption on a server, standard LUKS plus `dropbear-initramfs` remains the practical path: it lets you unlock the encrypted root remotely over SSH during boot instead of needing physical or console access on every restart.

### 2.2 Get used to living with sudo-rs

The default sudo implementation is now sudo-rs, a memory-safe Rust-based rewrite. It shows asterisks while you type the sudo password by default — a visible sign you're running the new version (disable via `Defaults !pwfeedback` if you prefer the old silent prompt). Test compatibility with any custom sudoers rules after upgrading.

### 2.3 Confirm TLS 1.0/1.1 is actually disabled

Apache and Nginx reject TLS 1.0/1.1 by default in 26.04. If you have legacy clients relying on those protocols, you'll need to explicitly re-enable them in your server config — but consider whether you actually need to.

### 2.4 Make use of post-quantum crypto being on by default

Post-quantum cryptography (ML-KEM, ML-DSA) is now the default in OpenSSL and OpenSSH 10.2 via hybrid algorithms. This protects against "harvest now, decrypt later" attacks where encrypted traffic is stored for future decryption. No action needed, but good to know when troubleshooting compatibility with older systems.

### 2.5 Harden SSH further

Even though OpenSSH 10.2 is more secure by default, you should still: - Disable password login: `PasswordAuthentication no` in `/etc/ssh/sshd_config` - Change the default port if the server is internet-facing - Use `AllowUsers` or `AllowGroups` to restrict who can log in

### 2.6 Set up the firewall with ufw

Example: - `sudo ufw allow OpenSSH` — allow SSH before you enable the firewall - `sudo ufw enable` - `sudo ufw status verbose` — see active rules

### 2.7 Install fail2ban to stop brute-force attempts

Example: `sudo apt install fail2ban && sudo systemctl enable --now fail2ban` — bans IP addresses that rack up repeated failed login attempts.

### 2.8 Explore AppArmor permission prompts

26.04 introduces experimental user prompts for snap application permissions via AppArmor. Worth keeping an eye on if you run graphical tools or snap-based services on the server.

### 2.9 Confirm SSSD no longer needs root

SSSD (the identity service) no longer runs with root privileges in 26.04, reducing the attack surface for LDAP/AD-integrated servers. Verify any custom SSSD configuration still works.

### 2.10 Put OpenLDAP in enforce mode deliberately

OpenLDAP runs in AppArmor enforce mode by default. If you use custom schemas or paths, test carefully in a non-production environment first so the AppArmor profile doesn't block legitimate traffic.

### 2.11 Keep the system patched automatically

Example: `sudo apt install unattended-upgrades` and configure `/etc/apt/apt.conf.d/50unattended-upgrades` to automatically install security updates.

### 2.12 Use Canonical Livepatch — now on Arm64 too

Livepatch applies kernel patches without a reboot and now supports Arm64 servers in addition to x86_64. Example: `sudo snap install canonical-livepatch && sudo canonical-livepatch enable <TOKEN>`

### 2.13 Protect yourself against legacy crypto

Deprecated DSA keys are fully removed in 26.04 (OpenSSH 10.0 dropped DSA support entirely, completing a deprecation that started when DSA was disabled by default back in OpenSSH 7.0/2015). Replace any remaining DSA-based SSH keys with Ed25519 or RSA before upgrading: `ssh-keygen -t ed25519 -C "server-key"`

### 2.14 Review file permissions regularly

Example: `find / -xdev -perm -4000 -type f 2>/dev/null` — find SUID files that can be a security risk if they're not actually needed.

---

## 3. Networking

### 3.1 Manage networking declaratively with Netplan

Example: edit `/etc/netplan/*.yaml` and run `sudo netplan apply` to apply changes without a reboot. Use `sudo netplan try` to test a config with automatic rollback if something goes wrong.

### 3.2 Switch to chrony for time sync

Chrony replaces systemd-timesyncd as the default in 26.04 for more accurate time synchronization. Example: `chronyc tracking` (current sync status), `chronyc sources -v` (list time servers).

### 3.3 Diagnose network problems fast

Example: - `ip addr show` — show interfaces and IP addresses - `ip route get 8.8.8.8` — see which route traffic to an address takes - `ss -tulpn` — show listening ports and processes

### 3.4 Set static DNS with systemd-resolved

Example: `resolvectl status` (show current per-interface DNS settings), `resolvectl dns eth0 1.1.1.1` (set a DNS server temporarily).

### 3.5 Test connectivity with modern tools

Example: `mtr example.com` — combines ping and traceroute in real time, better for troubleshooting intermittent network issues than separate tools.

### 3.6 Set up bonding/teaming for redundancy

On servers with multiple NICs, define a bond configuration in Netplan to improve availability against cable or port failures.

### 3.7 Monitor bandwidth usage live

Example: `sudo apt install nload && nload` — real-time graph of network traffic per interface.

---

## 4. Package Management & Updates

### 4.1 Get used to the APT 3 series

26.04 introduces APT 3, where `apt-key` is fully removed. Manage repository keys via `.gpg` files in `/etc/apt/keyrings/` instead of the old keyring system.

### 4.2 Keep the system clear of old packages

Example: - `sudo apt autoremove` — remove packages no longer needed - `sudo apt autoclean` — clear old package files from the cache - `sudo apt full-upgrade` — upgrade with smart dependency handling

### 4.3 Use snap deliberately on servers

Snap gives you sandboxed, self-updating packages — good for tools that need to stay current, but can mean higher memory usage. Example: `snap list` (see installed snaps), `sudo snap refresh` (update all).

### 4.4 Lock a package's version when needed

Example: `sudo apt-mark hold packagename` — prevent a specific package from being upgraded by accident (handy for, say, a custom kernel version). Undo with `sudo apt-mark unhold packagename`.

### 4.5 Check what an upgrade will actually do

Example: `apt list --upgradable` before `apt upgrade` to see exactly which packages are affected.

### 4.6 Find out where a command came from

Example: `dpkg -S $(which command)` — shows which package installed a given command, handy when troubleshooting conflicts.

### 4.7 Plan for the new split linux-firmware package

The formerly monolithic `linux-firmware` package (600 MB) is now split into 17 vendor-specific sub-packages. This reduces bandwidth on incremental updates, but check that the right sub-package for your hardware (graphics, wifi, network card) is installed after a minimal install.

### 4.8 Search package descriptions effectively

Example: `apt-cache search "text editor"` — find packages by function description when you can't remember the exact package name.

### 4.9 View detailed info about a package before installing

Example: `apt show packagename` — see version, size, dependencies, and description before installing anything on a production server.

### 4.10 Upgrade between LTS releases safely

Example: `sudo do-release-upgrade`. During remote upgrades the system automatically opens an extra SSH session on port 1022 as a safety net in case the normal connection drops — don't forget to temporarily allow that port in your firewall/security group.

---

## 5. systemd, Services & Processes

### 5.1 Analyze boot time

Example: `systemd-analyze blame` (show which services take longest at boot), `systemd-analyze critical-chain` (see the critical dependency chain).

### 5.2 Manage services efficiently

Example: - `systemctl status service` — status and recent log lines - `systemctl enable --now service` — enable and start in one step - `systemctl mask service` — prevent a service from being started by accident

### 5.3 Read logs smartly with journalctl

Example: - `journalctl -u service -f` — follow a service's logs live - `journalctl --since "1 hour ago"` — logs from the last hour - `journalctl -p err` — only error messages

### 5.4 Limit resources per service with cgroup v2

Since only cgroup v2 is supported in 26.04, use `systemctl edit service` and set `MemoryMax=`, `CPUQuota=` to limit per-service resource usage without external tools.

### 5.5 Create your own systemd timers instead of cron

Timers integrate better with logging and dependencies. Example: create a `.timer` and `.service` file, then enable with `systemctl enable --now my-job.timer`.

### 5.6 Check which services are listening on the network

Example: `sudo systemctl list-sockets` — show all socket-activated services.

### 5.7 Troubleshoot a service that won't start

Example: `systemctl status service -l --no-pager` followed by `journalctl -xeu service` for full failure detail.

### 5.8 Clean up old kernels automatically

Example: `sudo apt autoremove --purge` also removes old, unused kernel versions that would otherwise fill up `/boot` over time — check `dpkg -l | grep linux-image` if you're running low on space there.

---

## 6. Storage & File Systems

### 6.1 Take advantage of improved file cache handling

26.04 has improved support for F2FS, exFAT, and EXT4 with better file cache handling — good for servers with heavy I/O workloads like databases or log aggregation.

### 6.2 Monitor disk usage proactively

Example: `ncdu /` — an interactive, fast overview of what's using space (install with `sudo apt install ncdu`).

### 6.3 Set up ZFS for advanced storage

Example: `sudo apt install zfsutils-linux` and `sudo zpool create tank /dev/sdb` — ZFS gives you built-in snapshotting, compression, and data integrity.

### 6.4 Take instant snapshots with LVM

Example: `sudo lvcreate --size 5G --snapshot --name snap_backup /dev/vg0/root` — good for taking consistent backups without stopping services.

### 6.5 Check SMART status on physical disks

Example: `sudo apt install smartmontools && sudo smartctl -a /dev/sda` — catch early signs of disk failure before they become critical.

### 6.6 Automate disk cleanup with tmpfiles.d

Use `/etc/tmpfiles.d/` to automatically clear old temporary files and logs by rule, instead of manual cron jobs.

### 6.7 Extend an LVM volume with zero downtime

Example: `sudo lvextend -L +10G /dev/vg0/root && sudo resize2fs /dev/vg0/root` — grow storage live on a running server.

### 6.8 Check disk space per filesystem quickly

Example: `df -hT` — shows filesystem type alongside used/free space, handy for quickly spotting which partitions are approaching capacity.

---

## 7. Containers & Virtualization

### 7.1 Pick the right container track for your workload

26.04 introduces two tracks: the Application Track for teams who want regular feature updates, and the Stable Track for production that needs predictable component versions. Choose deliberately based on how much change your environment can tolerate.

### 7.2 Use Incus/LXD for lightweight containers

Example: `sudo snap install lxd && lxd init` — system containers with almost no overhead, a good alternative to full VMs for isolating services.

### 7.3 Run Docker on Ubuntu Server 26.04

Example: `curl -fsSL https://get.docker.com | sh` followed by `sudo usermod -aG docker $USER` to run without sudo.

### 7.4 Use ubuntu-virt-hwe for a newer virtualization stack

The new `ubuntu-virt-hwe` mechanism lets you pull a newer virtualization stack (QEMU/KVM/libvirt) from future interim releases while the rest of the system stays on the LTS base — useful if you need support for new hardware without leaving LTS.

### 7.5 Set up KVM virtualization

Example: `sudo apt install qemu-kvm libvirt-daemon-system virtinst bridge-utils` — full support for virtual machines with hardware acceleration.

### 7.6 Enable confidential computing when needed

26.04 has full support for Intel TDX and AMD SEV-SNP for memory encryption and integrity protection at the CPU level — relevant for multi-tenant environments with high security requirements.

### 7.7 Monitor container resources

Example: `docker stats`, or for LXD: `lxc info container-name --resources` — keep track of memory and CPU usage per container.

### 7.8 Build your own images with Packer

Example: combine cloud-init with HashiCorp Packer to build reusable, version-controlled Ubuntu 26.04 images for your infrastructure instead of configuring every instance by hand.

---

## 8. Cloud & Provisioning

### 8.1 Automate server configuration with cloud-init

Example: write a user-data file with packages, users, and commands to run on first boot — the same method works on AWS, Azure, GCP, Oracle Cloud, and IBM Cloud.

### 8.2 Check cloud-init status when troubleshooting

Example: `cloud-init status --long` and `sudo cat /var/log/cloud-init-output.log` — see exactly what happened during provisioning.

### 8.3 Use official cloud-optimized images

26.04 has ready-optimized images for AWS, Azure, Google Cloud, IBM Cloud, and Oracle Cloud — always start from these instead of building your own images from scratch when possible.

### 8.4 Be aware of the AMD64v3 requirement in the cloud

Every cloud provider's AMD64 images are now built with AMD64v3 as the baseline. Older instance types based on Intel Ivy Bridge or Sandy Bridge are no longer supported — check your instance type before upgrading.

### 8.5 Resize cloud instances safely

Before changing instance type, take a snapshot/image of the disk so you can roll back if the new instance type turns out to be missing support for something.

### 8.6 Manage multiple servers centrally with Landscape

Canonical Landscape gives you centralized patching, monitoring, and compliance reporting for fleets of Ubuntu servers — worth a look if you're managing more than a handful.

### 8.7 Tag and search cloud resources consistently

Use cloud-init metadata and your cloud provider's tagging system to label every 26.04 instance with owner, environment, and purpose — makes it much easier to find and clean up forgotten test instances.

---

## 9. Performance & Monitoring

### 9.1 Track system load in real time

Example: `htop` (install with `sudo apt install htop`) — interactive process monitoring, easier to read than `top`.

### 9.2 Analyze disk I/O per process

Example: `sudo apt install iotop && sudo iotop` — see exactly which processes are causing high disk load.

### 9.3 Collect historical performance data with sysstat

Example: `sudo apt install sysstat` and enable collection — `sar -u 1 5` shows CPU load over time, useful for capacity planning.

### 9.4 Use GPU compute straight from official sources

AMD ROCm joins NVIDIA CUDA in Ubuntu's official repositories for the first time in 26.04 (CUDA's own toolkit has been available there since 22.04 — ROCm is what's genuinely new). Example: `sudo apt install rocm-hip-sdk` for AMD, or `sudo apt install nvidia-cuda-toolkit` for NVIDIA — no third-party PPA needed for either.

### 9.5 Measure network latency against multiple targets at once

Example: `fping -a -g 192.168.1.0/24` — a fast sweep across an entire subnet to find live hosts.

### 9.6 Set up free monitoring with Netdata

Example: `sudo apt install netdata` — a real-time dashboard in your browser for CPU, memory, disk, and network, with no complex configuration.

### 9.7 Optimize for newer CPUs with x86-64-v3 package variants

If your hardware is from 2013 or later, look into the optional x86-64-v3 package variants, which can give performance gains through newer instruction sets.

---

## 10. Backup & Recovery

### 10.1 Run incremental backups with restic

Example: `restic backup /etc /home /var/www` against a local or cloud-based repository — fast, encrypted, and deduplicated.

### 10.2 Automate backup schedules with systemd timers

Combine restic or rsync with a systemd timer instead of cron for better logging and error handling of backup jobs.

### 10.3 Take consistent database backups

Example: `pg_dump -Fc mydatabase > backup.dump` for PostgreSQL 18 — use `-Fc` for a compressed format that supports selective restore.

### 10.4 Test your backups regularly

A backup that's never been restored isn't verified. Schedule regular restore tests in an isolated environment.

### 10.5 Sync files efficiently between servers

Example: `rsync -avz --delete /source/ user@server:/dest/` — `--delete` keeps the destination in sync by removing files no longer present in the source.

### 10.6 Use ZFS snapshots for fast recovery

If you're running ZFS: `sudo zfs snapshot tank@before-update` gives you an instant, near-zero-cost recovery point before risky changes.

---

## 11. Remote Access & Administration

### 11.1 Use tmux or screen for long-running sessions

Example: `tmux new -s my-job` — let processes keep running even if the SSH connection drops. Reconnect with `tmux attach -t my-job`.

### 11.2 Restrict root login over SSH

Set `PermitRootLogin no` in `/etc/ssh/sshd_config` and use sudo for administrative tasks instead — significantly reduces the attack surface.

### 11.3 Use passphrase-protected SSH keys with ssh-agent

Example: `ssh-add ~/.ssh/id_ed25519` — add your key to the agent once per session instead of typing the passphrase every time.

### 11.4 Set up MFA for sensitive servers

With the new authd framework, you can tie server login to cloud identity providers (Entra ID, Google IAM, OIDC), including multi-factor authentication.

### 11.5 Set up remote desktop securely when needed

If a graphical interface is required, tunnel over SSH rather than exposing VNC/RDP directly: `ssh -L 5900:localhost:5900 user@server`.

### 11.6 Automate configuration with Ansible

Example: write a playbook and run `ansible-playbook -i inventory.ini site.yml` to manage configuration idempotently across multiple servers.

---

## 12. Databases & Web Services

### 12.1 Run PostgreSQL 18 with modern features

26.04 ships PostgreSQL 18 by default. Example: `sudo apt install postgresql postgresql-contrib` and `sudo -u postgres psql` to connect.

### 12.2 Explore DocumentDB for document-based workloads

The new DocumentDB package provides MongoDB-like document storage on top of PostgreSQL — worth exploring if you need flexible, schema-less storage without introducing an entirely new database system.

### 12.3 Set up a LEMP stack

Example: `sudo apt install nginx mysql-server php-fpm` — a common foundation for web applications on Ubuntu Server.

### 12.4 Verify TLS configuration after installing

Since Apache and Nginx now reject TLS 1.0/1.1 by default, test your configuration with `curl -v --tlsv1.2 https://your-server` to confirm modern clients still connect correctly.

### 12.5 Use Certbot for free TLS certificates

Example: `sudo apt install certbot python3-certbot-nginx && sudo certbot --nginx` — automatic certificate management and renewal via Let's Encrypt.

### 12.6 Monitor database performance

Example: enable `pg_stat_statements` in PostgreSQL to identify slow queries directly in the database instead of guessing.

---

## 13. Ubuntu Pro, WSL & More

### 13.1 Register the server for Ubuntu Pro

Example: `sudo pro attach <TOKEN>` — gives access to Expanded Security Maintenance (ESM), Livepatch, and compliance tooling (CIS, FIPS), free for up to 5 personal machines.

### 13.2 Check which Pro services are active

Example: `pro status --all` — a clear overview of which security services are enabled or available.

### 13.3 Run Ubuntu Server tools in WSL for development

26.04 has improved WSL integration with cloud-init and Ubuntu Pro, making it easier to mirror a production server's configuration locally during development on Windows.

### 13.4 Keep an eye on RISC-V and Arm support

If you're experimenting with RISC-V hardware, remember that only RVA23S64-based boards are supported in 26.04 (currently mostly via QEMU) — older RVA20 boards won't work.

### 13.5 Update Raspberry Pi firmware before installing

Raspberry Pi 4 requires boot firmware dated November 25, 2022 or later. Update via `raspi-config` on an older install before switching to 26.04.

### 13.6 Check the official release notes when in doubt

Before major production changes, always double-check `documentation.ubuntu.com/release-notes/26.04/`, since details on known issues and package-specific changes are updated on an ongoing basis.

### 13.7 Stick to LTS-to-LTS upgrades in production

Don't hop through interim releases (like 25.10) on production servers unless you actively need a specific feature from one — stick to LTS-to-LTS for maximum stability and support length.

---

*© SplitZrate 2026 · 100+ Tips & Tricks — Ubuntu Server 26.04 LTS*