

What is the best Linux?

“ I want to learn Linux because I’m a web developer and build WordPress websites. But in the future, I also want to develop actual apps, which is why I’d like to learn Node.js and React, for example. Anyway, I’m not quite sure yet which Linux version is “good”—for example, Centmin Mod requires Rocky Linux or AlmaLinux, while others think Debian is a good choice (because it’s stable), and still others are Ubuntu fans. So which Linux version should I learn, and do I even need to commit to one? Isn’t it enough to “learn Linux once,” or are the differences significant?

The practical answer: **learn “Linux,” pick one daily driver, stay flexible** ☐

You *don’t* need to permanently commit to a single distribution to “learn Linux.” Most of what you’ll use as a web developer—filesystem layout, permissions, processes, SSH, networking basics, logs, system services—transfers extremely well across distros.

What *does* vary meaningfully is mostly:

- **Package management** (APT vs DNF/YUM, etc.)
- **Release model** (LTS/stable vs faster-moving)
- **Defaults & tooling** (firewall tooling, SELinux/AppArmor, paths/config conventions)
- **Ecosystem compatibility** (what your hosting/platform or tools expect)

So the best approach is: **master the shared core**, then become comfortable translating the “last mile” differences.

What you should learn first (the cross-distro core) ☐☐

If you learn these well, you can work on almost any Linux server:

1. Shell basics

- navigating (`cd`, `ls`, `find`)
- text processing (`grep`, `sed`, `awk`, `less`)
- pipes/redirection (`|`, `>`, `2>`)

2. Filesystem & permissions

- ownership, `chmod`, `chown`, groups
- symlinks, `umask`

3. Users, SSH, keys

- `ssh`, `sshd`, `~/.ssh/authorized_keys`

4. Processes & resource troubleshooting

- `ps`, `top/htop`, `systemctl`, `journalctl`

5. Services & boot

- **systemd** basics (common on most modern distros)

6. Networking

- `ss`, `curl`, DNS basics, firewall basics

7. Logs

- `/var/log`, `journald`, app logs (Nginx/Apache, PHP-FPM)

8. Web stack basics

- Nginx/Apache configs, TLS (Let's Encrypt), reverse proxying

That's "Linux once." The rest is "Linux per environment."

Which distro should *you* learn? Pick based on where you'll run things

If you want a solid "server-first" baseline for WordPress hosting

Debian is an excellent default choice:

- Very stable, widely used on servers
- Great documentation and community
- APT is straightforward
- Easy to host WordPress stacks reliably

Ubuntu Server LTS is also a great pick:

- Very common in cloud providers and tutorials
- Good hardware/cloud support
- Faster access to newer packages than Debian stable (but still “LTS stable”)

Rule of thumb:

- If you like “boring and stable”: **Debian**
- If you like “popular and well-supported by tutorials/cloud”: **Ubuntu LTS**

If your tooling *requires* RHEL-like distros (Rocky/Alma)

If you specifically want **Centmin Mod**, then yes: learn **Rocky Linux** or **AlmaLinux** (RHEL-family).

- You’ll use **DNF/YUM**
- You’ll run into **SELinux** more often (important in real server ops)
- Very common in enterprise-like environments

But note: this is often more about matching a specific stack/tool than being “better Linux.”

Do the differences matter day-to-day? Some do—most don’t

Here are the differences you’ll actually *feel*:

1. **Package manager & package names**
 - Debian/Ubuntu: `apt`, `apt-get`
 - Rocky/Alma: `dnf` (formerly `yum`)
2. **Security framework defaults**
 - Ubuntu often emphasizes **AppArmor**

- RHEL family commonly uses **SELinux**

3. **Release cadence**

- Debian stable: slower, very stable
- Ubuntu LTS: stable but more current
- RHEL family: stability + long lifecycle

4. **Some config/layout conventions**

- Usually minor; most key paths are similar, but defaults and helper tools vary

What's largely the same:

- `systemd`, `sshd`, file permissions, standard GNU tooling, TCP/IP basics, Nginx/PHP/MySQL concepts.
-

A good strategy for you (web dev → WordPress → Node/React) ☐☐

Recommended path

1. **Choose one “daily driver” distro to learn deeply**

- I'd pick **Ubuntu LTS** or **Debian**.

2. **Use Docker for app/dev workflows**

- Node/React dev environments become reproducible across distros and even macOS/Windows.

3. **Add a second distro later for “translation skill”**

- Once comfortable, spin up a **Rocky/Alma** VM and practice the same tasks with `dnf` + SELinux basics.

This mirrors real life: you'll often develop locally (maybe Ubuntu/Debian), but deploy to whatever the server/host requires.

Concrete recommendation (if you want me to pick)

- **Learn on Ubuntu Server LTS** (most “how-to” coverage, cloud-friendly) or **Debian** (excellent stability culture).

- **Don't stress about committing forever.** After you learn the core, switching is mostly learning a new package manager + a few defaults.
 - If **Centmin Mod** is a serious near-term goal, add **Rocky/Alma** sooner rather than later.
-

If you tell me these 3 things, I'll tailor the best choice

1. Are you mostly using **VPS**, shared hosting, or local dev?
2. Do you plan to manage your own server stack (Nginx/PHP/MySQL) or use panels/tools?
3. Are you on Windows/macOS/Linux for your main dev machine?

Answer those, and I'll recommend a setup (distro + learning plan) that matches your WordPress and future Node/React goals.

Revision #1

Created 2026-04-04 19:17:10 UTC by art10m

Updated 2026-04-04 19:19:44 UTC by art10m