

Linux System Administration

Course Duration ~70 Hours · 10 Weeks · Live Online · Lab-Heavy · You Break Systems Deliberately and Recover Them

Curriculum reviewed: July 2026 — the enterprise distributions and their certification framework changed recently; this syllabus is checked against vendor documentation

[Shell](#)[systemd](#)[Storage](#)[Networking](#)[Hardening](#)[Ansible](#)[Image Mode](#)[Troubleshooting](#)

In short: Everything runs on Linux — learn to configure it, secure it, automate it, and to be the person who can still fix it when the dashboard is red and nobody knows why. This is the **foundation course of the whole DevOps, Cloud & Security track**: the layer underneath Docker, Kubernetes, Terraform, CI/CD and security work, which all assume this knowledge and none of which teach it. It is reachable with **no programming background at all** — the widest entry point in the section — and it is built around the one thing free tutorials never cover: **diagnosis under pressure**. On the honest question of whether "system administrator" is still a career: the job title is fading, the skill is now compulsory, and it appears in far more job descriptions than the title does — people who skip these fundamentals can follow a runbook but can't diagnose anything it doesn't already cover. Troubleshooting is the spine, not an appendix: from Module 4 onward you deliberately break systems and recover them. Modern surface throughout — systemd as the centre of the system, mandatory access control taught as something to **debug** rather than disable, container tooling on the host, the newer packet-filtering framework, Ansible-based configuration as code, post-quantum-ready cryptography, and **image mode**: the operating system built and shipped as a bootable container image with atomic updates and rollback. On AI: enterprise Linux now ships an assistant that suggests packages and fixes — a suggestion is a hypothesis, not an answer, so you verify it against the manual pages, the package database and the system's actual state before running it as root, and the leading exams are performance-based with no assistance. Aligned to the leading vendor-specific enterprise Linux administrator credential (whose framework was restructured in 2026 into specialisation tracks and progressive levels) and the main vendor-neutral alternatives. 10 weeks, ~70 hours, lab-heavy.

Duration	10 weeks · ~70 hours · live + hands-on	Format	Live online · lab-heavy · you will break systems deliberately and recover them
Prerequisite	General computer literacy. No programming, scripting or Linux experience required — everything taught from first principles. The widest entry point in the section.	Curriculum	10 modules, from the shell to a full break-fix and hardening capstone
Certification	Aligned to the leading vendor-specific enterprise Linux administrator credential and the main vendor-neutral alternatives	Ideal for	Fresh graduates from any background, support/helpdesk staff, Windows admins adding Linux, and developers who deploy to Linux but can't debug it

Is "System Administrator" Still a Career? An Honest Answer

The job title is fading; the skill is now compulsory. Every container runs on a Linux kernel, every cloud instance is a Linux box, every pipeline executes on a Linux runner, and every production incident eventually ends at a shell prompt. DevOps, SRE, cloud, platform and security roles all assume this knowledge and none of them teach it — so the skill appears in far more job descriptions than the title does. People who skip these fundamentals hit a ceiling fast: they can follow a runbook but can't diagnose anything the runbook doesn't already cover. Free tutorials teach commands; they don't teach method — how to reason from symptom to cause, under time pressure, on a system you didn't build, while something is down. That is the spine of this course.

Working With AI on a Production System — Without Breaking It

Enterprise Linux now ships with an AI assistant that suggests packages, configurations and troubleshooting steps in plain language. Used well it is a genuine accelerator; used carelessly it is a fast way to break production as root. The discipline we teach is simple and explicit: a suggestion is a hypothesis, not an answer — you verify it against the manual pages, the package database and the system's actual state, and you understand what it does, before you run it. The leading certification exams are performance-based with no external assistance, so the underlying knowledge is not optional; the AI can't sit the exam or carry the 2 a.m. incident for you.

Where This Course Sits in the Track

- **Docker & Kubernetes** owns container orchestration — here you learn container fundamentals on the host (runtime, rootless/daemonless operation, host container tooling) and stop there
- **Cybersecurity** owns the security discipline — here you learn host hardening (mandatory access control, firewalling, auditing, patching) as system administration
- **Terraform** owns provisioning the machine — this course owns what the provisioned machine actually does once it exists

Course Highlights

- Every module leads with what you configure, diagnose or recover — not what you "learn about"
- Troubleshooting as the spine — a systematic diagnostic method you apply from Module 4 onward
- No programming background needed — the widest entry point into the entire DevOps track
- systemd as the centre of the system — services, targets, timers, cgroups and the journal
- Mandatory access control taught as something to debug, not disable — the field's most common bad habit, corrected
- Image mode — the OS built and shipped as a bootable container image, with atomic updates and rollback
- Configuration as code with Ansible — administer many machines without logging into any of them
- A timed break-fix gauntlet and a capstone that takes an unmanaged environment to hardened, automated and documented

Hands-on Projects & Portfolio

- **Multi-Server Lab Build** — from installation to a hardened, monitored baseline across several machines
- **Storage Extend & Recover** — extend a filesystem under a running service, then recover a host that won't boot after a bad mount
- **systemd Deep Lab** — write custom units and timers, then diagnose and repair a set of broken services with no hints
- **Network Services Build-Out** — core services with firewalling, TLS and access control
- **Hardening Against a Benchmark** — before/after evidence and a written justification for every exception
- **Access-Control Workshop** — debug real mandatory-access-control denials and write correct policy rather than turning it off
- **Ansible Automation** — convert a manual server build into a repeatable, idempotent playbook
- **Image Mode Exercise** — build a bootable OS image with container tooling, deploy it, update it atomically and roll it back
- **Break-Fix Gauntlet** — timed scenarios: a full disk, a DNS failure, a permissions problem, a runaway process, a failed boot
- **Capstone** — an unmanaged environment taken to hardened, automated, documented and defensible

Tools & Stack

Linux (RHEL-family & Debian-family) · Bash · systemd · SSH · LVM · SELinux · firewalld · nftables · DNF/APT · Podman · bootc image mode · Ansible · cron & systemd timers · rsync · Git · Vim · tcpdump · Wireshark · Prometheus · Grafana · journald · auditd · KVM · Docker

Certification Alignment

- Mapped to the **leading vendor-specific enterprise Linux administrator certification** and the main vendor-neutral alternatives (such as the Linux Professional Institute certifications and CompTIA Linux+)
- The vendor's certification framework was **restructured in 2026** into specialisation tracks and progressive levels — an optional entry-level tier, a baseline administrator credential, and engineer and architect levels within each track. Much of the older path advice online is now out of date; we align to the current structure, not the retired ladder
- The leading exam is **fully performance-based** — you fix real systems under time pressure rather than answer questions, which is exactly why this course is built around labs
- Presented as preparation and alignment only — no official partnership or accreditation implied, and no exam outcome guaranteed

Career Outcomes

Linux administration is less a destination than a prerequisite — the substrate under containers, cloud, automation and security work, which is why the skill appears in far more job descriptions than the job title does. It remains one of the most accessible technical entry points for people without

a programming background, particularly from support, networking and operations roles, and the enterprise distributions dominate in banking, telecom, government and large services organisations. Roles: Linux System Administrator · DevOps Engineer · Site Reliability Engineer · Cloud Engineer · Platform Engineer · Infrastructure Engineer · Support & NOC Engineer · Security Engineer · Database or Application Administrator on Linux platforms.

Who This Course Is For

- **Final-year students and fresh graduates from any background** who want a technical entry point that does not require programming
- IT support, helpdesk and desktop professionals moving into infrastructure
- Windows administrators adding Linux
- **Developers who deploy to Linux but can't debug it** — a very common and quietly limiting gap
- Aspiring DevOps, cloud and SRE engineers who skipped the fundamentals and hit a ceiling
- Security professionals who need real host-level depth; network engineers expanding into systems
- Anyone preparing for a performance-based Linux certification

Module 1: Foundations & the Shell

7 Hours

YOU CONFIGURE

- Your own virtualised lab, and enough shell fluency that the command line stops being an obstacle and becomes the fastest way to get anything done

TOPICS

- Distribution families and which one you'll actually meet at work; installation and virtualisation for your lab; the filesystem hierarchy
- Navigating and manipulating files; the shell environment; redirection and pipes; text processing with the classic tools; regular expressions; editors
- Getting fluent enough that the command line stops being an obstacle

Assessment: Quiz + Lab

Module 2: Users, Permissions & Access

6 Hours

YOU CONFIGURE

- Accounts, groups and least-privilege access on a real system — including key-based SSH that you've hardened yourself

TOPICS

- Accounts, groups and the password and shadow files; standard and special permissions; access control lists
- Privilege escalation with sudo and sane sudoers policy; SSH key-based authentication and hardening; session management
- The principle of least privilege applied to real systems

Assessment: Quiz + Lab

Module 3: Package & Software Management

6 Hours

YOU CONFIGURE

- Repositories, a patch strategy and a clean rollback path — so software changes are deliberate and reversible, not hopeful

TOPICS

- Repositories and package managers; dependency resolution; modules and application streams
- Sandboxed application packaging; building and installing from source when you must
- Patch management strategy; rollback

Assessment: Quiz + Lab

Module 4: systemd & Process Management

8 Hours

YOU DIAGNOSE

- A failed service — methodically, from the journal to the unit file to the root cause — instead of guessing. This is where the course's troubleshooting spine begins

TOPICS

- Units, services, targets, timers, mounts and user units; writing unit files from scratch; dependencies and ordering
- Resource control with control groups; the boot process end to end; changing the default target
- The journal and log management; diagnosing a failed service methodically rather than by guesswork

Assessment: Quiz + Lab**Module 5: Storage**

7 Hours

YOU RECOVER

- A system that will not boot because of a storage change — after learning to resize volumes safely under a running service

TOPICS

- Partitioning; filesystems and their trade-offs; mounting and persistent mount configuration; logical volume management including resizing
- Swap; RAID concepts; quotas; network and automounted filesystems; encrypted volumes; snapshots
- Recovering a system that will not boot because of a storage change

Assessment: Quiz + Lab**Module 6: Networking**

7 Hours

YOU DIAGNOSE

- A network failure end to end — because it is always DNS, until you can prove it isn't — and stand up the common services behind a firewall

TOPICS

- Addressing and routing; modern network configuration tooling; DNS resolution and why it is always DNS
- SSH in depth including tunnelling; firewalling with the current framework and the newer packet filtering layer
- Network troubleshooting methodology; common services such as web, file and time servers

Assessment: Quiz + Lab**Module 7: Security & Hardening**

8 Hours

YOU HARDEN

- A host to a published benchmark with evidence for every control — and you debug a mandatory-access-control denial instead of disabling enforcement, the field's most common bad habit

TOPICS

- Mandatory access control: understanding contexts, booleans and how to debug a denial instead of disabling enforcement; auditing and file integrity monitoring
- Host hardening baselines and benchmarks; intrusion detection basics; certificate and cryptography management including post-quantum-ready algorithms
- Compliance profiles; a documented hardening exercise

Assessment: Quiz + Lab**Module 8: Automation & Configuration as Code**

8 Hours

YOU AUTOMATE

- A manual server build into a repeatable, idempotent playbook — and manage a fleet without logging into any single machine

TOPICS

- Shell scripting properly: variables, conditionals, loops, functions, argument handling, error handling and exit codes; scheduled work with timers

- Ansible for declarative configuration — inventories, playbooks, roles, idempotency, vendor-supplied system roles
- Managing many machines without logging into any of them

Assessment: Quiz + Lab

Module 9: Containers & Image Mode

7 Hours

YOU BUILD

- A bootable, immutable operating system as a container image — deploy it, update it atomically and roll it back — the convergence point between traditional administration and container practice

TOPICS

- Containers on the host; rootless and daemonless operation; images and registries; running containers as managed services
- Image mode: building and deploying a bootable immutable operating system using container tooling, with atomic updates and rollback
- When a traditionally package-managed host is still the right answer

Assessment: Quiz + Lab

Module 10: Performance, Troubleshooting & Capstone

8 Hours

YOU DELIVER

- A deliberately misconfigured environment made correct, hardened, automated and documented — after proving you can diagnose a saturated system and recover an unbootable one under time pressure

TOPICS

- The systematic diagnostic method; CPU, memory, I/O and network analysis; identifying resource saturation; tracing and observability tools; log correlation
- Kernel parameters and tuning; backup and restore strategy; disaster recovery; rescue and single-user boot; root password recovery; documenting an incident
- Capstone: take a deliberately misconfigured environment and make it correct, hardened, automated and documented

Assessment: Capstone Project + Break-Fix Gauntlet