

Foundations, Linux, Bash, and Git

4-week detailed curriculum | Target: Linux Essentials (010-160)

I built this guide to give you a clear path into technology without expecting you to already know the language. You are not supposed to understand everything on day one. Your job is to show up consistently, practice, ask questions, and keep evidence of what you build. I will help you connect the pieces, but the hands-on work is where the confidence will come from.

Study rule: Do not only watch videos. Type the commands, build the labs, break things, fix them, and explain what happened in your own words.

Your outcome for this phase

Build the computer, command-line, troubleshooting, and version-control foundation required for every later phase.

Expected pace: 2-3 hours Monday through Friday, 3-4 hours Saturday, and Sunday reserved for rest or catch-up. Each week ends with something you can demonstrate.

Phase learning resources

Resource	How to use it
LPI Linux Essentials learning materials	Official objective-aligned study material.
Linux Journey	Beginner-friendly lessons on commands, permissions, processes, and networking.
OverTheWire Bandit	Command-line challenges that make Linux practice feel like a game.
GitHub Skills	Interactive Git and GitHub exercises inside real repositories.
Learn Git Branching	Visual, interactive Git branching practice.

How to prove you learned it

- Explain the concept without reading notes.
- Complete the lab twice: once with instructions and once from memory.
- Push scripts, diagrams, screenshots, or written reflections to GitHub.
- Write down one failure, the cause, and the fix each week.
- Answer practice questions only after performing hands-on work.

Week 1 - How computers and operating systems work

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Computer components: CPU, RAM, storage, motherboard, firmware; explain what each part does.	Notes + commands/screenshots
Tue	Operating systems, kernels, applications, processes, and services; compare Windows and Linux.	Notes + commands/screenshots
Wed	Install VirtualBox or use WSL; create Ubuntu; take a snapshot; document the setup.	Notes + commands/screenshots
Thu	Filesystem concepts: root, home, paths, extensions, hidden files, and mounting.	Notes + commands/screenshots
Fri	Basic terminal navigation: pwd, ls, cd, mkdir, touch, cp, mv, rm; complete 20 command repetitions.	Notes + commands/screenshots
Sat	Mini-lab: create a folder hierarchy for a fictional company and write a one-page summary of how a computer starts.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 2 - Linux command line and text processing

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Read files with cat, less, head, tail; redirect output with > and >>.	Notes + commands/screenshots
Tue	Pipes and search: , grep, find, locate, wc, sort, uniq.	Notes + commands/screenshots
Wed	Text editing with nano or vim; create and edit configuration files.	Notes + commands/screenshots
Thu	Archives and downloads: tar, gzip, zip, unzip, wget, curl.	Notes + commands/screenshots
Fri	Environment variables, PATH, shell history, aliases, quoting, wildcards.	Notes + commands/screenshots
Sat	Lab: OverTheWire Bandit levels 0-10; save commands and lessons in GitHub notes.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 3 - Users, permissions, processes, and services

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Users and groups: /etc/passwd, useradd, passwd, groups, sudo.	Notes + commands/screenshots
Tue	Permissions: rwx, chmod numeric and symbolic modes, chown, umask.	Notes + commands/screenshots
Wed	Processes: ps, top, jobs, bg, fg, kill, signals, process IDs.	Notes + commands/screenshots
Thu	Services and logs: systemctl, journalctl, /var/log, service status.	Notes + commands/screenshots
Fri	Packages: apt update, install, remove, repositories, package search.	Notes + commands/screenshots
Sat	Lab: create two users, shared group access, install NGINX, inspect logs, and troubleshoot a stopped service.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 4 - Bash scripting, Git, and exam review

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Bash scripts, shebang, variables, read, echo, exit codes.	Notes + commands/screenshots
Tue	Conditionals, loops, functions, positional parameters, basic error handling.	Notes + commands/screenshots
Wed	Git lifecycle: init, status, add, commit, log, diff, .gitignore.	Notes + commands/screenshots
Thu	Branches, merge, clone, pull, push, and pull requests.	Notes + commands/screenshots
Fri	Build Linux Administration Toolkit: disk check, service check, log archive, user report.	Notes + commands/screenshots
Sat	Linux Essentials objective review, practice questions, weak-area lab, and exam readiness decision.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Final phase assessment

Do not move forward based only on time spent. Move forward when you can perform and explain the following:

- Complete the main project without copying every step.
- Troubleshoot at least three intentionally introduced failures.
- Explain the architecture or workflow using a diagram.
- Score at least 80% on two separate practice assessments when a certification is targeted.
- Teach the core concepts back to another person in plain language.

Certification note

Target: **Linux Essentials (010-160)**. Certification dates should be adjusted based on readiness. Passing quickly is less important than retaining the skill and being able to use it in a real environment.