

Sprint 1 – Development Environment and DevOps Basics

Hatim Salhi

February 9, 2026

Contents

1	Development Environment	2
1.1	Terminal and Shell	2
1.2	Code Editors	2
2	Linux Basics	2
2.1	Filesystem Structure	2
2.2	Essential Commands	2
2.3	Command-line Editors	2
3	Git and Version Control	2
3.1	Fundamental Concepts	2
3.2	Practice	3
4	Web and DevOps Culture	3
4.1	Web Servers	3
4.2	Web Architecture	3
4.3	Security and Hosting	3
5	Conclusion	3

1 Development Environment

1.1 Terminal and Shell

- **iTerm2 (macOS) / Windows Terminal (Windows):** modern terminal for productivity.
- **Zsh + Oh My Zsh:**
 - Configured with auto-completion, aliases, and themes.
 - Fast directory navigation and command execution.
- **Tmux:**
 - Manage multiple sessions in the same terminal.
 - Split screen to run multiple commands simultaneously.
- **Result:** ready-to-code environment with command history and project management.

1.2 Code Editors

- **VS Code:** optimized with extensions for PHP, Python, Laravel, and Git.
- Keybindings, themes, and integrated terminal configured.
- Goal: lightweight, powerful IDE ready for all projects.

2 Linux Basics

2.1 Filesystem Structure

- Main directories: `/home`, `/etc`, `/usr`, `/var`, `/tmp`.
- Understanding permissions and users.

2.2 Essential Commands

- Navigation and file management: `ls`, `cd`, `mv`, `cp`, `rm`.
- Permissions: `chmod`, `chown`.
- Searching and filtering: `grep`, `find`.
- Viewing files: `cat`, `less`, `head`, `tail`.
- Automation: bash scripts and task scheduling with `cron`.

2.3 Command-line Editors

- **nano:** simple editor for quick file changes.
- **vim:** powerful editor with insert and command modes.

3 Git and Version Control

3.1 Fundamental Concepts

- Repositories, commits, branches, merges, rebases.
- History tracking and collaboration.

3.2 Practice

- Usage via terminal.
- Exercises on Learn Git Branching.
- Creating and merging branches to simulate real workflows.

4 Web and DevOps Culture

4.1 Web Servers

- Roles of Apache and Nginx servers.
- Interaction with clients over HTTP/HTTPS.
- Managing ports and network services.

4.2 Web Architecture

- Frontend vs backend.
- Database, application server, and web server.

4.3 Security and Hosting

- SSL/TLS certificates to secure connections.
- SSH connections for secure remote administration.
- Best practices: permissions, SSH keys, firewalls.

5 Conclusion

- The development environment is ready for efficient coding.
- Linux and Git fundamentals enable effective project management and collaboration.
- Web and DevOps concepts provide a clear understanding of infrastructure.
- Next steps: practical application with Laravel, Docker, and CI/CD pipelines.