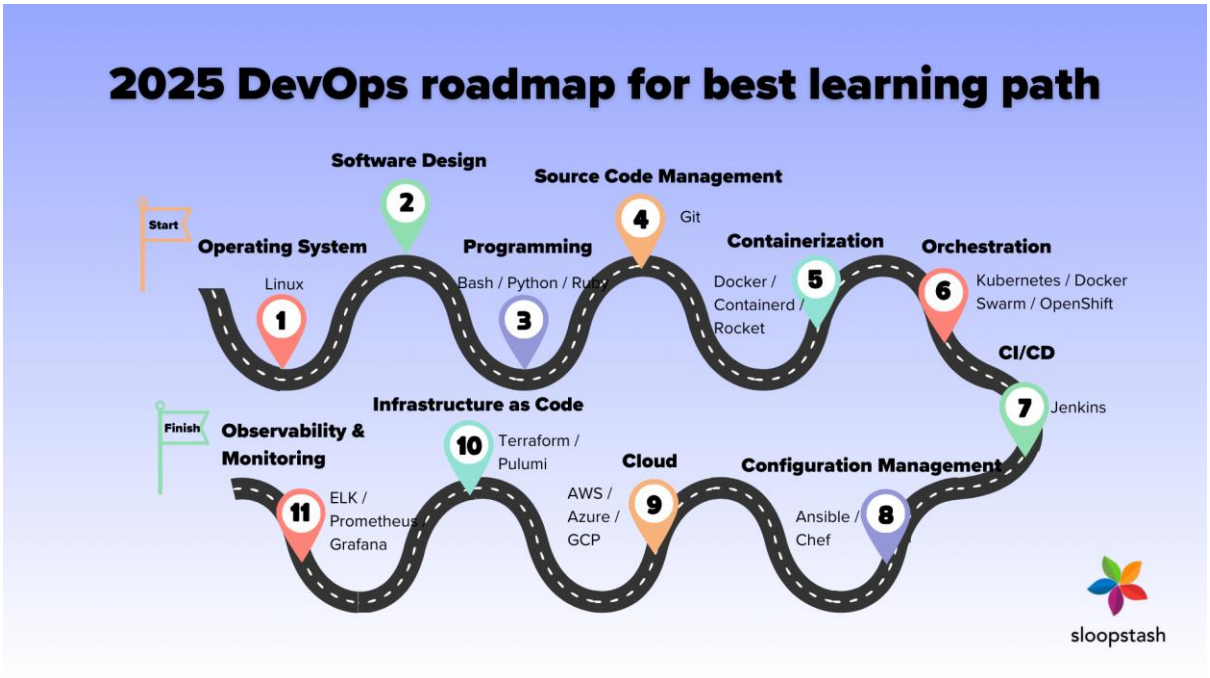
The Complete DevOps Roadmap



Introduction

If you want to start a career in DevOps or level up your existing skills, you are in the right place. DevOps is not just a job title.It is a culture and a set of practices that bridges the gap between software development and operations. A DevOps engineer ensures that applications are built, tested, deployed, monitored, and scaled reliably.

In this article, you will find a step-by-step roadmap with the essential skills, tools, and technologies you need to master to become a successful DevOps engineer.

1. Master the Basics of Linux

Linux is the backbone of servers and cloud environments. As a DevOps engineer, you will use the command line extensively to configure systems and deploy applications.

What to learn:
File system navigation (ls, cd, pwd, find)
File operations (cp, mv, rm, touch, cat)
User and permissions management (chmod, chown)
Process management (ps, top, kill)
Package managers (apt, yum, dnf)
Bash scripting basics

Resources:
https://linuxjourney.com/
The Linux Command Line by William Shotts

2. Networking Fundamentals

Networking connects everything—servers, applications, and users. You need to know how data flows and how to troubleshoot issues.

What to learn:
OSI and TCP/IP models
IP addressing, subnetting, DNS, DHCP
Common protocols: HTTP/HTTPS, FTP, SSH
Firewalls and security groups
Tools: ping, traceroute, netstat, Wireshark

Resources:
https://inl.info.ucl.ac.be
https://www.wireshark.org/docs/

3. Git and Version Control

Git is the backbone of collaboration in DevOps. Every workflow starts with version control.

What to learn:
Git basics: clone, init, commit, push, pull
Branching and merging strategies
Resolving merge conflicts
Working with remote repositories (GitHub, GitLab, Bitbucket)

Resources:
Pro Git Book
https://docs.github.com/en/get-started/quickstart/hello-world

4. Programming Basics (Python Recommended)

Programming is essential for automation in DevOps. Python is recommended for its simplicity and versatility.

What to learn:
Python syntax, variables, and data structures (lists, dicts, sets, tuples)
File handling and error handling
Modules and packages
Automation scripts for deployment and log parsing

Resources:
https://automatetheboringstuff.com/

5. Cloud Computing

Cloud platforms are central to modern DevOps. Focus on one cloud provider, such as AWS.

What to learn:
Launch and manage virtual servers
Storage services and IAM (users, roles, policies)
Virtual private clouds and network management
Managed services like databases and serverless computing

Resources:
https://aws.amazon.com/free/
https://aws.amazon.com/certification/certified-cloud-practitioner/

6. Containers (Docker)

Containers make applications portable and consistent across environments.

What to learn:
Build Docker images
Run and manage containers
Write Dockerfiles
Multi-container apps with Docker Compose

Resources:
https://labs.play-with-docker.com/
Docker Deep Dive by Nigel Poulton

7. CI/CD Pipelines

Continuous Integration and Continuous Delivery ensure faster and safer software releases.

What to learn:
Jenkins basics: jobs, pipelines, Jenkinsfiles
Integrating Git with Jenkins
Running automated tests
Deploying applications automatically after successful builds

Resources:
https://www.jenkins.io/doc/
https://docs.github.com/en/actions

8. Container Orchestration (Kubernetes)

Kubernetes is the industry standard for managing containerized applications at scale.

What to learn:
Kubernetes architecture (master node, worker nodes)
Core concepts: pods, deployments, services, namespaces
Scaling applications
Networking model in Kubernetes
Helm basics for package management

Resources:
https://kubernetes.io/docs/home/
https://www.manning.com/books/kubernetes-in-action

9. Web Servers and Infrastructure Services

Understanding how applications are served to users is crucial.

What to learn:
Nginx as reverse proxy and load balancer
Forward proxy configuration
Caching strategies
Firewalls and security groups

Resources:
https://nginx.org/en/docs/

10. Configuration Management

Configuration management automates deployment, configuration, and management of servers and applications.

What to learn:
Ansible basics: playbooks, roles, modules
Managing variables and templates
Automating deployment across environments

Resources:
https://docs.ansible.com/
https://www.ansiblefordevops.com/

11. Infrastructure as Code (IaC)

IaC allows you to define infrastructure through code, making it repeatable and version-controlled.

What to learn:
Terraform basics: configuration files, providers, modules
Remote states and workspaces
Managing cloud resources using code

Resources:
https://developer.hashicorp.com/terraform/docs

12. Monitoring and Logging

Monitoring ensures system reliability, and logging helps troubleshoot issues.

What to learn:
Prometheus for metrics collection
Grafana for visualization and alerts
ELK Stack for centralized logging

Resources:
https://prometheus.io/docs/
https://grafana.com/tutorials/

Final Thoughts

Becoming a DevOps engineer requires consistent practice and hands-on experience. Build projects, deploy apps to the cloud, break things, and fix them. Focus on learning by doing, and gradually integrate the tools and practices outlined in this roadmap.