DevOps Mastery

Phase 1: Foundation (Months 1-2.5)

Month 1: DevOps Culture & Linux Deep Dive

DevOps Culture (Crucial Context)

Books:

- **The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win** (432 pages)
  - Duration: 10 days (approx. 43 pages/day)
  - Goal: Understand the "why," cultural shifts, and the core problems DevOps solves
- **The DevOps Handbook** (480 pages)
  - Duration: 10 days (focus on Part II: Technical Practices & Part III: The Three Ways)
  - Note: Skim remaining sections
  - Goal: Get foundational understanding of key DevOps practices and principles

Linux Mastery

Core Book:

- **Linux Command Line and Shell Scripting Bible** (816 pages)
  - Duration: 10 days
  - Focus: Chapters 4-20 (commands, scripting, process, permissions, basic networking commands)
  - Daily: 80 pages/day of targeted reading + 2 hours hands-on practice

Practice Resource:

- **Linux Journey** (Home | Linux Journey)
  - Daily: 30-60 mins for interactive tutorials
  - Goal: Become highly comfortable and efficient on the Linux command line

Month 2: Networking Deep Dive & Git Proficiency

Networking Fundamentals (In-depth)

Core Book:

- **Computer Networking: A Top Down Approach, 7th** (converted.pdf)
  - Duration: 15 days
  - Focus: Chapters 1-5 (Application, Transport, Network, Data Link layers)
  - Note: Skip overly detailed physical layer
  - Daily: 40-50 pages + 1 hour practical lab (VM networking, ping, traceroute, netstat, curl usage)

AWS Documentation:

- **AWS VPC Documentation** (docs.aws.amazon.com/vpc/)
  - Duration: 15 days
  - Focus: VPCs, subnets, route tables, security groups, NACLs, internet gateways, NAT gateways, peering
  - Include hands-on labs
  - Goal: Understand how applications communicate across networks and within cloud environments

Git Proficiency (Crucial for CI/CD)

Core Book:

- **Pro Git** (456 pages - free online)
  - Duration: 10 days
  - Focus: Chapters 1-7 (core concepts, branching strategies, rebase, cherry-pick, remote operations)
  - Daily: ~45 pages + 1.5-2 hours of deliberate Git practice
  - Practice: Complex branching, conflict resolution, reflog, Git hooks basics
  - Goal: Master essential Git commands and advanced workflows for collaborative development and CI/CD

Month 2.5: Infrastructure as Code (Terraform & AWS CDK)

IaC Principles & Terraform

Core Book:

- **Terraform: Up & Running** (352 pages)
  - Duration: 12 days (approx. 29 pages/day)
  - Goal: Understand core Terraform concepts (providers, resources, data sources, state, modules)

Documentation:

- **HashiCorp Terraform Documentation** (Terraform overview | Terraform | HashiCorp Developer)
  - Duration: Continuous focus on AWS Provider
  - Daily: 1-2 hours hands-on labs building AWS resources (VPC, EC2, S3, DynamoDB, Lambda, API Gateway)

AWS CDK

Books:

- **AWS CDK Dev Guide 2025.pdf**
- **Gupta A. Mastering Infrastructure as Code with AWS CloudFormation...2025.pdf**
  - Duration: 8 days total
  - Focus: Advanced patterns and CDK's relationship with CloudFormation

Project: Convert existing AWS CDK projects to Terraform for comparison

- Goal: Be proficient in defining and managing AWS infrastructure using both Terraform and AWS CDK

Phase 2: Containerization & Orchestration (Months 3-5)

Month 3: Docker Mastery

Core Book:

- **Docker Deep Dive** (368 pages)
  - Duration: 12 days (approx. 31 pages/day)
  - Goal: Master Docker concepts (images, containers, volumes, networks, Docker Compose)

Documentation:

- **Docker Documentation** (Home)
  - Duration: 18 days
  - Focus: Dockerfile best practices, multi-stage builds, networking, basic container security
  - Daily: 1.5-2 hours hands-on practice

Project: Containerize Django and React applications, optimizing for image size and build time

Month 4: Configuration Management (Ansible)

Core Book:

- **Ansible for DevOps** (464 pages)
  - Duration: 15 days (approx. 31 pages/day)
  - Goal: Understand Ansible core concepts (playbooks, modules, roles, inventory, variables)

Documentation:

- **Ansible Documentation** (Ansible Documentation)
  - Duration: 15 days
  - Focus: Specific modules, roles, and creating reusable playbooks
  - Daily: 1.5-2 hours hands-on practice (automate server setup, deploy applications, manage services)

Project: Automate server configuration for applications (install dependencies, configure Nginx, deploy application files)

Month 5: Kubernetes Fundamentals

Core Book:

- **Kubernetes in Action** (624 pages)
  - Duration: 20 days (approx. 31 pages/day)
  - Focus: Practical chapters (Pods, Deployments, Services, ConfigMaps, Secrets, Volumes, Ingress basics)
  - Note: Skim deep theoretical sections
  - Goal: Build solid understanding of Kubernetes core concepts

Documentation:

- **Kubernetes Official Documentation** (Kubernetes Documentation)
  - Duration: 10 days with intense hands-on practice
  - Tools: Start with Minikube/Kind locally, then move to AWS EKS
  - Daily: 2 hours of kubectl commands, YAML manifest creation, and deployment

Project: Deploy containerized applications to Kubernetes cluster (Minikube → EKS on AWS)

Phase 3: CI/CD, Observability & Security (Months 6-8)

Month 6: CI/CD Implementation (GitHub Actions) & Nginx

CI/CD Principles

Core Book:

- **Continuous Delivery – Reliable Software Releases Through Build, Test And Deployment Automation.pdf** (500 pages)
  - Duration: 15 days
  - Focus: Principles of automated building, testing, and deployment
  - Note: Skim implementation details of older tools
  - Goal: Understand strategic importance and mechanics of effective CI/CD

CI/CD Tool

Documentation:

- **GitHub Actions Documentation** (GitHub Actions documentation - GitHub Docs)
  - Duration: 10 days with intensive hands-on practice
  - Daily: 2 hours creating and refining workflows for Django/React projects

    - Focus: Build, test, push Docker images to ECR, deploy to EC2/EKS

Web Servers (Nginx)

Documentation:

- **Nginx Documentation** (nginx documentation)
    - Duration: 5 days
    - Focus: Reverse proxy, load balancing, static file serving, SSL/TLS termination, basic configuration
    - Daily: Hands-on setup of Nginx as reverse proxy for Django app

Project: Build complete CI/CD pipeline using GitHub Actions. Deploy application behind Nginx reverse proxy

Month 7: Monitoring & Observability

Prometheus

Documentation:

- **Prometheus Documentation** (Overview | Prometheus)
    - Duration: 15 days
    - Focus: Installation, configuration, PromQL, Alertmanager
    - Daily: 1.5-2 hours hands-on setup (node exporter, application metrics)

Grafana

Documentation:

- **Grafana Documentation** (Technical documentation | Grafana Labs)
    - Duration: 15 days
    - Focus: Data sources, dashboard creation, panel types, alerting
    - Daily: 1.5-2 hours hands-on building dashboards for metrics

AWS CloudWatch

Documentation:

- **AWS CloudWatch Documentation** (docs.aws.amazon.com/cloudwatch/)
    - Duration: Integrate as needed for AWS-native logging and metrics

Project: Implement comprehensive monitoring stack for deployed applications, visualizing key metrics and setting up alerts

Month 8: Security (DevSecOps) & Secrets Management

Security Fundamentals

Core Book:

- **Securing DevOps: Security in the Cloud** (384 pages)
    - Duration: 15 days (approx. 25 pages/day)
    - Focus: Practical security measures, shifting left, security in cloud context
    - Goal: Understand how to integrate security throughout DevOps pipeline

Documentation:

- **OWASP Top 10** (OWASP Top Ten | OWASP Foundation)
    - Duration: Study common web application vulnerabilities and mitigation

Secrets Management

Documentation:

- **AWS Secrets Manager** (docs.aws.amazon.com/secretsmanager/)
- **HashiCorp Vault** (Vault product documentation | Vault | HashiCorp Developer)
  - Duration: 15 days (alternating focus)
  - Focus: Secure storage, access control, secret rotation, integration with applications/CI/CD

Project: Integrate secrets management into applications and CI/CD pipelines. Implement basic security scanning in CI pipeline (e.g., using Bandit for Python)

Phase 4: Advanced Cloud, Automation & Mastery (Months 9-10)

Month 9: Advanced AWS & Python Automation

Advanced AWS Patterns

Books:

- **AWS-Serverless-Applications-Lens.pdf**
  - Duration: 10 days
  - Focus: Scalable serverless patterns
- **Kumar A. Ultimate AWS CDK for Infrastructure Automation...2025.pdf**
  - Duration: 10 days
  - Focus: Advanced CDK patterns and real-world use cases

Python for DevOps

Core Book:

- **Python for DevOps** (544 pages)
  - Duration: 10 days
  - Focus: Automation scripts, Boto3 integration, cloud APIs, custom monitoring tools
  - Daily: 50 pages + 1.5 hours hands-on Python scripting

AWS Well-Architected Framework

Documentation:

- **AWS Well-Architected Framework** (AWS Well-Architected - Build secure, efficient cloud applications)
  - Duration: Continuous review
  - Focus: Security, Reliability, Performance Efficiency, Cost Optimization pillars

Project: Build Python automation scripts for cost reporting, security group auditing, or deployment management specific to AWS environment

Month 10: Scalability & Capstone Project

System Design & Scalability

Core Book:

- **Designing Data Intensive Applications** (550 pages)
  - Duration: 15 days
  - Focus: Chapters 1-6 (Foundations of Data Systems: Reliable, Scalable, Maintainable)
  - Note: Skim later chapters unless directly relevant
  - Goal: Invaluable architectural context for building scalable systems

MLOps/Advanced Pipelines

Book:

- **Testas A. Building Scalable Deep Learning Pipelines on AWS...2024.pdf**
    - Duration: 5 days
    - Focus: DevOps/MLOps aspects of pipeline and infrastructure design

Advanced Container Orchestration

Documentation:

- **AWS ECS/EKS Advanced Features** (docs.aws.amazon.com/ecs/, docs.aws.amazon.com/eks/)
    - Duration: 5 days
    - Focus: Service discovery, auto-scaling, advanced security, integration with Load Balancers, Route 53

Capstone Project

Duration: 5 days (dedicated, intense work)

Goal: Deploy complete, production-ready application stack from scratch (portfolio piece)

Components:

- Terraform/CDK for IaC
- GitHub Actions for CI/CD
- Docker for containerization
- Kubernetes/ECS for orchestration
- Prometheus/Grafana/CloudWatch for monitoring
- Nginx for ingress/load balancing
- Integrated security/secrets management

Objective: Complex project showcasing mastery of all learned concepts and problem-solving abilities

---

Books Needed:

1. Linux Command Line and Shell Scripting Bible (816 pages)
2. The Phoenix Project (432 pages)
3. The DevOps Handbook (480 pages)
4. Pro Git (456 pages)
5. Terraform: Up & Running (352 pages)
6. Ansible for DevOps (464 pages)
7. Docker Deep Dive (368 pages)
8. Kubernetes in Action (624 pages)
9. Securing DevOps (384 pages)
10. Python for DevOps (544 pages)
