Full DevOps Starter Kit

What's inside: - 30-Day Daily Study Plan (detailed) - Visual roadmap (text form) - Tools & learning order - Daily checklist you can follow - Portfolio projects (with brief steps) - Free resources & course suggestions - Beginner DevOps resume tips

---

1. Quick Start (How to use this kit)

1. Follow the **30-day schedule** day-by-day; do hands-on practice each day.
2. Keep a single GitHub repo called `devops-playground` and push notes and code.
3. For each completed task, tick it in the **Daily Checklist** section.
4. Use the Resources section to pick one course/video per topic.

---

2. 30-Day Daily Study Plan (1.5–2 hrs/day)

Week 1 — DevOps basics, Linux, Git

Day 1 — 2: DevOps overview, CI/CD concepts, DevOps lifecycle, key terms (CI, CD, IaC, monitoring) — 30m theory + 1h notes/practice.

Day 3 — 4: Install Ubuntu (VM or WSL). Linux basics: `ls`, `cd`, `pwd`, `mkdir`, `rm`, `cp`, `mv`, `cat`, `tail`, `head`, `grep`, `find`, `chmod`, `chown`.

Day 5 — 6: Processes (`ps`, `top`), background jobs, `ssh`, users/groups, basic networking (`ifconfig`/`ip`, `netstat`), package managers.

Day 7: Git & GitHub basics: clone, commit, branch, merge, pull requests. Create repo and push notes.

Week 2 — Cloud Fundamentals (AWS suggested)

Day 8 — 9: Cloud fundamentals & IAM (users, roles, policies). Learn console and CLI basics.

Day 10 — 11: EC2: launch instance, connect via SSH, basic web server (nginx/apache) deploy a static page.

Day 12: S3: create bucket, upload objects, static site hosting basics, permissions.

Day 13: Networking basics: VPC, subnets, security groups, NACLs (high level).

Day 14: Mini project: Host static assets on S3; serve a small site from EC2 that uses S3 for images.

Week 3 — CI/CD (Jenkins / GitHub Actions)

Day 15 — 16: Jenkins install (or use GitHub Actions): pipelines vs jobs, pipeline concepts.

Day 17 — 18: Create a pipeline that pulls code from GitHub and runs tests (or a build script).

Day 19 — 20: Write a `Jenkinsfile` or GitHub Actions workflow file: build → test → package.

Day 21: CI mini project: On git push, run pipeline that builds and archives an artifact and posts a status.

Week 4 — Containers & Orchestration

Day 22 — 23: Docker fundamentals: images, containers, Dockerfile, `docker run`, volumes, networking.

Day 24 — 25: Docker Compose: define multi-container app (app + DB). Dockerize a simple web app.

Day 26 — 27: Kubernetes basics with Minikube or Kind: pods, deployments, services, namespaces.

Day 28 — 29: Ingress, scaling (HPA basics), config maps and secrets.

Day 30: Final project: Dockerize app, deploy to Kubernetes cluster, expose service and test scaling.

---

3. Roadmap (From zero → intermediate)

Foundations → Source Control → CI/CD → Containers → Orchestration → Infrastructure as Code → Monitoring & Logging

Text roadmap (progress checkpoints): - Foundations: Linux + scripting (Bash) - Source control: Git + GitHub workflows - CI/CD: Jenkins / GitHub Actions / GitLab CI - Containerization: Docker + Docker Compose - Orchestration: Kubernetes basics → Helm - IaC: Terraform → module development → remote state - Monitoring & Logging: Prometheus + Grafana, ELK stack - Security & Observability: Secrets management, tracing

---

4. Tools & Suggested Learning Order

1. Linux (Ubuntu/WSL)
2. Git & GitHub
3. Docker (images, containers, volumes)
4. CI tool (Jenkins or GitHub Actions)

5. Kubernetes (Minikube / Kind)
6. Terraform (IaC)
7. Cloud provider (AWS recommended): EC2, S3, IAM
8. Monitoring: Prometheus + Grafana
9. Logging: ELK or hosted alternatives

---

5. Daily Checklist (copy and use)

- [ ] Read theory for the topic (20 min)
- [ ] Hands-on practice / labs (60 min)
- [ ] Write notes to GitHub repo (20 min)
- [ ] Quick review / plan tomorrow (10 min)

---

6. Portfolio Projects (resume-ready)

Project A — CI/CD pipeline for sample app - Tools: GitHub + Jenkins or GitHub Actions + EC2 - Steps: create app repo → write pipeline → build & deploy to EC2 → document steps

Project B — Dockerized fullstack app - Tools: Docker, Docker Compose - Steps: containerize frontend & backend → compose file → local testing → push image to Docker Hub

Project C — Kubernetes deployment - Tools: Kubernetes, Minikube - Steps: create manifests (Deployment, Service, Ingress) → deploy → test rolling updates

Project D — IaC automation - Tools: Terraform + AWS - Steps: declare VPC + EC2 + S3 → apply → destroy → use remote state

Project E — Monitoring dashboard - Tools: Prometheus + Grafana - Steps: instrument app metrics → scrape with Prometheus → visualize in Grafana

---

7. Free Resources & Course Suggestions

(Choose one resource per topic to avoid overload) - Linux basics: free YouTube playlists (search "Linux for Beginners") - Git & GitHub: Git official docs + free GitHub Learning Lab - Docker: Docker docs + free tutorials on YouTube - Kubernetes: Kubernetes official docs + "Kubernetes by Example" tutorials - CI/CD: Jenkins official docs or GitHub Actions docs - Terraform: HashiCorp Learn (official) - AWS basics: AWS Free Tier + AWS Cloud Practitioner learning path - Monitoring: Prometheus and Grafana official docs

---

8. Beginner DevOps Resume Tips

- Add a “Projects” section with 3–4 portfolio projects linking to GitHub repos.
- Mention tools you used and what you automated (e.g., "Automated CI pipeline using Jenkins to build and deploy Node app to EC2").
- Keep each bullet outcome-focused: what you built, what it improved (deploy time, reliability).
- Add a short summary: "Software engineer transitioning to DevOps — skilled in CI/CD, Docker, basic AWS, and Terraform."

---

9. Extra: Weekly Progress Tracker (copyable table)

Week	Goals	Completed (Y/N)	Notes
1	Linux + Git basics		
2	AWS basics + small deploy		
3	CI pipeline working		
4	Docker & Kubernetes deployed		

---

10. Next actions I can help with right now

- Generate a downloadable PDF (I can create it here)
- Export the kit as a markdown or DOCX file
- Create a one-page visual roadmap image
- Build a daily checklist file (CSV/MD)

End of document.