

DevOps: Docker, Kubernetes & CI/CD

Take an application from a Git repository to a monitored Kubernetes deployment with automated pipelines

Cloud and DevOps · intermediate · Live online, instructor-led

This syllabus is a draft awaiting instructor review and may change.

A 56-hour live course in practical DevOps: Linux, Git, Bash scripting, Docker and Compose, CI/CD with GitHub Actions, Kubernetes and Helm, infrastructure-as-code with Terraform, monitoring with Prometheus and Grafana, and a capstone pipeline.

Length and schedule

56 hours of live teaching, on either schedule:

- Weekday Intensive (Mon–Fri): about 12 weeks before holidays
- Weekend Evening (Sat & Sun evenings): about 14 weeks before holidays

Fees per student, whole course

- Group: Rs. 16,999
- Duo: Rs. 27,999
- One-to-One: Rs. 49,999

Launch fee — GST treatment to be confirmed.

What you will be able to do

- Work confidently on a Linux server: files, permissions, processes, services, networking and logs
- Collaborate with Git using branches, pull requests and a clear branching strategy
- Automate routine tasks with safe, readable Bash scripts
- Containerise applications with efficient, secure Dockerfiles and run multi-service stacks with Docker Compose
- Build CI/CD pipelines in GitHub Actions that lint, test, scan, build and publish container images
- Deploy and operate applications on Kubernetes with Deployments, Services, Ingress, ConfigMaps, Secrets, probes and Helm
- Describe infrastructure as code with Terraform and explain state, plans and modules
- Monitor applications with Prometheus and Grafana, set alerts and choose a safe deployment strategy

Before you start

- Comfortable with basic terminal commands: navigating folders, running programs and editing files
- Basic programming in any language: you can read a short script or small application
- A general idea of how a web application talks to an API and a database
- A laptop with at least 8 GB RAM (16 GB recommended for Docker and a local Kubernetes cluster) running Linux, macOS, or Windows with WSL 2
- A free GitHub account for the Git and CI/CD modules

Tools

Ubuntu Linux (or WSL 2 on Windows), Bash and common command-line tools, Git and GitHub, Docker Engine or Docker Desktop, Docker Compose, GitHub Actions, GitHub Container Registry, Trivy, Kubernetes with kind, kubectl, Helm, Terraform, Prometheus, Grafana, Visual Studio Code

Syllabus

Module 1: Linux for DevOps

The Linux skills every later module depends on: the shell, permissions, services, networking and logs.

1. The Linux filesystem and the shell (2 h)
2. Users, permissions, processes and services (2 h)
3. Networking tools and SSH (2 h)
4. Packages, logs and Linux lab with doubt-clearing (2 h)

Module 2: Git and collaboration

Version everything and work on shared code through branches and reviewed pull requests.

5. Git fundamentals and history (2 h)
6. Branching, pull requests and doubt-clearing (2 h)

Module 3: Scripting and automation

Automate repetitive work with Bash scripts that are safe to run more than once.

7. Bash scripting basics (2 h)
8. Robust scripts, scheduling and doubt-clearing (2 h)

Module 4: Containers with Docker

Package applications into images that run the same way on every machine.

9. Containers and images (2 h)
10. Writing Dockerfiles (2 h)
11. Volumes, networks and Docker Compose (2 h)
12. Image security, registries and doubt-clearing (2 h)

Module 5: Continuous integration and delivery

Build, test and publish every change automatically, and release it safely.

13. CI/CD concepts and GitHub Actions (2 h)
14. A build and test pipeline (2 h)
15. Building and publishing images (2 h)
16. Deployment strategies, mid-course review and doubt-clearing (2 h)

Module 6: Kubernetes fundamentals

Deploy, configure, expose and update containerised applications on Kubernetes.

17. Kubernetes architecture and kubectl (2 h)
18. Pods, Deployments and rolling updates (2 h)
19. Services and Ingress (2 h)
20. Configuration, Secrets, probes and resources (2 h)
21. Helm and Kubernetes lab with doubt-clearing (2 h)

Module 7: Infrastructure as code

Describe infrastructure in reviewed code instead of clicking through consoles.

22. Infrastructure-as-code concepts and Terraform basics (2 h)

23. State, modules and doubt-clearing (2 h)

Module 8: Monitoring and observability

Know when an application is unhealthy before users tell you, and know what to do next.

24. Metrics with Prometheus and dashboards with Grafana (2 h)

25. Logs, alerts and doubt-clearing (2 h)

Module 9: Capstone: delivery pipeline

Plan, build and present a complete pipeline from commit to monitored deployment.

26. Capstone planning and build clinic (2 h)

27. Capstone peer review (2 h)

28. Final presentations and final assessment (2 h)

Final project

Build a complete delivery pipeline for a small web application. You receive an application with an API, a web front end and a PostgreSQL database, plus its source code in a Git repository. You deliver:- A short delivery plan: how changes move from a branch to production, what each pipeline stage checks, and how you would roll back.- Production-quality Dockerfiles (multi-stage, non-root, small images) and a Docker Compose file for local development.- A GitHub Actions workflow that lints and tests on every pull request, scans the images with Trivy, and builds, tags and publishes them to GitHub Container Registry on merge.- A Helm chart that deploys the application to Kubernetes with Deployments, Services, an Ingress, ConfigMaps, Secrets, resource limits and readiness and liveness probes; the pipeline deploys it to a kind cluster and runs a smoke test.- Terraform code that creates part of the supporting infrastructure, with a plan reviewed in the pull request.- Prometheus metrics and a Grafana dashboard for the application, with at least one alert rule, plus a short runbook for that alert.- A Bash script that automates one operational task, such as a database backup or an environment check. Reviewed live in the final week: you demonstrate a change flowing through the pipeline to the cluster, trigger and explain your alert, show a rollback, and answer questions about your design choices.

Certificate

- At least 80% attendance
- All assignments submitted
- Projects passed
- Final assessment passed (60%)

A ZORK Academy course completion certificate; paying for the course does not by itself earn it.

Next step

Book a free consultation: <https://zorkacademy.online/consultation?course=devops-docker-kubernetes>

We help you prepare for work; we do not guarantee jobs, placements or salaries.