CLOUD, DEVOPS, AND CONTAINERS

DevOps Foundations

Level: Foundation • 2 days (expandable to 3) • Virtual, In-person

Overview

DevOps has been mainstream for well over a decade, and yet most organizations still ship slowly, deploy nervously, and keep development and operations pointing fingers at each other. That is because DevOps is routinely mistaken for a toolchain or a job title, when it is actually a way of organizing work: shortening the path from idea to running software, and learning from what happens in production.

This is a hands-on, foundation course. Rather than catalog every DevOps tool, it teaches a smaller set of ideas deeply: the principles of flow, feedback, and continuous learning, and the core practices (version control for everything, continuous integration, continuous delivery, and monitoring) that put them into effect. The gradient runs from why DevOps exists, through the delivery pipeline step by step, to measurement and how change actually takes hold. Every module includes a lab, and each module builds on the one before.

Who Should Attend

- Developers, testers, and operations staff new to DevOps practices
- Team leads and managers whose teams are adopting DevOps
- Anyone who needs a shared vocabulary for how modern software gets delivered

Prerequisites

- Familiarity with how software is built and released in some organization
- No programming or tool experience required

What You Will Learn

- Explain what DevOps is, what problems it solves, and what it is not
- Describe flow, feedback, and continuous learning, and spot where a delivery process violates them
- Map a value stream and find the real bottleneck in a release process
- Explain the delivery pipeline: version control, continuous integration, continuous delivery, and deployment
- Apply the DORA metrics to measure delivery performance honestly
- Judge where a DevOps improvement should start in your own organization

Course Outline

Day one: the ideas and the flow of work

- Why DevOps Exists
  - The wall between development and operations, and what it costs
  - Flow, feedback, and continuous learning: the three ideas underneath everything
  - What DevOps is not: a team name, a toolchain, a job title
  - Lab: diagnose a fictional organization's painful release and name what is going wrong

- Seeing the Work: Value Streams
  - From idea to production: mapping how work actually moves
  - Wait time versus work time, and why the bottleneck is rarely where people think
  - Lab: map the value stream for a release process and locate its constraint
- Version Control and Continuous Integration
  - Everything in version control: code, configuration, infrastructure
  - Continuous integration: small changes, automated tests, fast feedback
  - Lab: walk a change through a real CI pipeline and read what it reports

Day two: delivery, feedback, and making it stick

- Continuous Delivery and Deployment
  - The deployment pipeline: from commit to production-ready
  - Deployment strategies in plain language: blue-green, canary, and feature flags
  - Lab: extend the pipeline so a change deploys to a test environment automatically
- Feedback from Production
  - Monitoring, logging, and alerting: knowing what the software is doing
  - Blameless postmortems and learning from failure
  - Lab: use monitoring data to find the cause of a simulated incident
- Measuring and Improving
  - The DORA metrics: what good delivery performance looks like
  - Culture and change: why DevOps adoptions fail, and what works instead
  - Where your organization should start
  - Lab: assess a team against the DORA metrics and propose its first improvement

Extended Version

The three-day version keeps the same gradient and adds:

- Deeper hands-on pipeline work, connecting to *CI/CD with GitHub Actions*
- Infrastructure as code and its role in DevOps, previewing *Infrastructure as Code with Terraform*
- DevSecOps: building security into the pipeline rather than bolting it on
- A capstone: design a DevOps improvement roadmap for a realistic organization and defend it