

DEVOPS

ADVANCED

5 DAYS

Platform Engineering and Internal DevOps Platform Bootcamp

Build a strong foundation in Internal Developer Platform (IDP) engineering, from golden paths to self-service automation and governance. Learn how to standardize delivery with GitOps, infrastructure automation, baseline observability, and scalable workflows that improve developer productivity and platform reliability.

Why Platform Engineering and IDP?

Faster developer delivery

Standard templates and automation reduce time-to-first-deploy and speed up releases.

Consistency across teams

Shared standards for deployment, monitoring, and security reduce variability and outages.

Lower operational load

Less ticket-driven operations through self-service workflows and paved roads.

Scalable modernization

A platform enables microservices and Kubernetes adoption without increasing cognitive load.

Secure and governed deployments

GitOps and IaC increase auditability, consistency, and change control.

What you'll be able to do

- ✓ Understand platform engineering fundamentals and how it differs from traditional DevOps operating models.
- ✓ Explain what an Internal Developer Platform (IDP) is and why modern organizations build one.
- ✓ Identify the key problems IDPs solve including developer friction, inconsistent environments, slow releases, repeated manual work, and lack of governance.
- ✓ Build an IDP foundation using golden paths, self-service workflows, standard templates, and opinionated infrastructure.
- ✓ Implement standardized developer workflows on Kubernetes including deployments, ingress routing, config and secrets patterns, autoscaling concepts, and resource limits.
- ✓ Design and implement GitOps-based delivery with pull-based deployments, environment consistency, and drift control.
- ✓ Automate infrastructure using Terraform including modules and reuse, environment separation, and secure state handling.
- ✓ Implement observability and reliability foundations including metrics and logs overview and SLO thinking for platform services.
- ✓ Design platform governance including RBAC and access boundaries, policy guardrails concepts, and cost control principles.
- ✓ Deliver a platform capstone that integrates Kubernetes, GitOps, CI pipelines, and a self-service new-service bootstrap workflow.

Who this is for

- Platform Engineers
- DevOps Engineers
- SRE Teams
- Cloud Engineers
- Engineering Productivity Teams
- Solution Architects and Tech Leads
- Engineering Managers driving developer experience improvements

Curriculum

01 Platform Engineering Fundamentals (Modern Operating Model)

- What platform engineering is and why it emerged
- Platform engineering vs traditional DevOps operating models
- Product mindset for platforms (platform as a product)
- Platform team roles, responsibilities, and success metrics
- **Hands-on:** Activity: Identify friction points in a typical delivery workflow and map them to platform opportunities

02 Internal Developer Platform (IDP) Concepts and Value

- What an IDP is and key capabilities (self-service, standards, automation)
- Why organizations build IDPs (speed, consistency, governance)
- Core building blocks (templates, workflows, runtime platform, policy, observability)
- Build vs buy considerations (Backstage concept overview)
- **Hands-on:** Workshop: Define the MVP scope of an IDP for a company with multiple engineering teams

03 Problems IDPs Solve (Developer Experience and Governance)

- Reducing developer friction and onboarding time
- Eliminating inconsistent environments and manual setup
- Accelerating releases and reducing delivery bottlenecks
- Improving governance without slowing teams down
- **Hands-on:** Activity: Create a “top 10 friction list” and design how an IDP removes each pain point

04 Golden Paths and Self-Service Foundations (IDP Base Layer)

- Golden paths concept (standardized happy-path workflows)
- Self-service workflows (create service, deploy, add config, scale)
- Standard templates for services (repo templates, deployment templates)
- Opinionated infrastructure and standards (defaults that reduce choices)
- **Hands-on:** Lab: Design a golden path for “create a new microservice and deploy to Kubernetes”

05 Standardized Kubernetes Developer Workflows (Production-Aligned)

- Standard workload setup (Deployment, Service, Ingress)
- Config and secrets patterns (ConfigMaps, Secrets, safe injection)
- Autoscaling concepts (HPA) and reliability-aware scaling
- Resource requests/limits and safe baseline sizing patterns
- **Hands-on:** Lab: Deploy a service with Ingress, config patterns, probes, autoscaling, and resource limits

06 GitOps Delivery for IDPs (Pull-Based Deployments and Drift Control)

- Why GitOps fits platform engineering (consistency, auditability)
- Repo structure patterns for environments (dev/stage/prod)
- Promotion workflow via pull request and approvals
- Drift detection and self-healing concepts
- **Hands-on:** Lab: Implement GitOps-based deployment and validate drift detection/self-heal behavior

07 Infrastructure Automation with Terraform (Modules, Environments, State)

- Terraform fundamentals for platform foundations
- Module design and reuse patterns (standard infra building blocks)
- Environment separation (dev/stage/prod) and variable strategy
- Secure state handling (remote state, locking, access boundaries)
- **Hands-on:** Lab: Build reusable Terraform modules and provision environment-specific infrastructure safely

08 Observability and Reliability Foundations (Platform SLO Thinking)

- Platform observability basics (metrics and logs overview)
- SLO mindset for platform services (availability, latency, error rates)
- Monitoring developer workflows (deploy success rate, MTTR, drift events)
- Reliability practices (runbooks, alert hygiene concepts)
- **Hands-on:** Workshop: Define SLOs and dashboards for core platform capabilities and golden paths

09 Governance and Cost Control for Platforms (RBAC and Guardrails)

- RBAC and access boundaries for teams and namespaces
- Policy guardrails concepts (admission controls, allowed images, security defaults)
- Cost control principles (requests/limits, quotas concepts)
- Auditability and compliance readiness for platform actions
- **Hands-on:** Activity: Design a governance model for platform usage with roles, guardrails, and cost controls

10 Capstone Project (IDP Foundation with Self-Service Bootstrap)

- Capstone goal: Deliver an IDP-style workflow integrating Kubernetes, GitOps, and CI
- Bootstrap a new service using a standard template (golden path)
- Automate deployment using CI pipeline + GitOps promotion workflow
- Provision required infrastructure via Terraform modules (optional environment separation)
- Add observability signals and governance guardrails
- **Hands-on:** Capstone Lab: Demo new-service creation → deploy → promote → observe with documentation and runbook notes