

DEVOPS

INTERMEDIATE

3 DAYS

FinOps for DevOps Engineers Bootcamp

Build a strong foundation in FinOps practices for DevOps teams, from cost visibility to optimization and governance. Learn how to reduce cloud waste using right-sizing, scheduling, autoscaling, and guardrails in IaC and CI/CD while building cost accountability through dashboards, budgets, and operational controls.

Why FinOps?

Reduced cloud waste

Identify idle resources, right-size workloads, and optimize spending.

Improved forecasting

Predict cloud spend trends and plan budgets with real usage insights.

Better cost visibility

Tagging, budgeting, and reporting improves accountability across teams.

More profitable cloud adoption

Align engineering decisions with business outcomes and ROI.

Smarter scaling decisions

Balance performance and cost using measurable unit economics.

What you'll be able to do

- ✓ Understand what FinOps is and why it matters for engineering teams.
- ✓ Identify common cloud cost drivers across compute, storage, networking, and managed services.
- ✓ Implement cost visibility using tagging/labeling standards and cost allocation models.
- ✓ Build engineering-led cost optimization processes such as right-sizing, autoscaling, scheduling idle resources, and waste elimination.
- ✓ Optimize Kubernetes spend using resource requests/limits, bin packing concepts, and over-provisioning detection.
- ✓ Implement cost guardrails in CI/CD with IaC review, cost estimation concepts, policy checks, and approvals.
- ✓ Build cost dashboards and reporting workflows for engineering teams.
- ✓ Establish governance with budgets, alerts, anomaly detection, and accountability per team/service.
- ✓ Create a practical FinOps implementation plan for DevOps organizations.

Who this is for

- DevOps Engineers
- Platform Engineers
- SRE Teams
- Cloud Engineers
- Engineering Managers who own cloud spend
- FinOps teams working closely with engineering

Curriculum

01 FinOps Fundamentals (Why Engineering Teams Must Care)

- What FinOps is and how it differs from traditional cost management
- FinOps principles (collaboration, ownership, continuous optimization)
- Engineering accountability for cloud cost and performance
- Key FinOps metrics (unit cost, cost per environment/service, waste)
- **Hands-on:** Activity: Identify top cost pain points in a sample cloud architecture

02 Cloud Cost Drivers (Compute, Storage, Network, Managed Services)

- Compute drivers (instance sizing, idle runtime, autoscaling gaps)
- Storage drivers (hot vs cold, snapshots, growth, retention)
- Network drivers (egress charges, cross-zone/cross-region traffic)
- Managed services drivers (databases, caching, logs/metrics costs)
- **Hands-on:** Lab: Review a sample billing breakdown and classify costs by driver category

03 Cost Visibility (Tagging/Labeling and Cost Allocation)

- Why visibility comes before optimization
- Tagging/labeling standards (env, owner, team, service, cost-center)
- Cost allocation models (shared vs dedicated, chargeback/showback concepts)
- Tracking cost by service and environment over time
- **Hands-on:** Lab: Define a tagging standard and map spend to team/service using sample data

04 Engineering-Led Optimization Processes (Right-Sizing and Waste Elimination)

- Right-sizing fundamentals (CPU/memory actual usage vs provisioned)
- Autoscaling patterns and ensuring scaling is effective
- Scheduling idle resources (non-prod shutdown patterns)
- Waste elimination checklist (unused disks, snapshots, orphaned resources)
- **Hands-on:** Workshop: Create an optimization backlog with impact/effort prioritization

05 Kubernetes Cost Optimization (Requests/Limits, Bin Packing, Over-Provisioning)

- How Kubernetes spend happens (nodes, pods, over-provisioning)
- Requests and limits impact on scheduling and cost
- Bin packing concepts and node utilization patterns
- Over-provisioning detection (idle capacity, wasted reservations)
- **Hands-on:** Lab: Tune workload requests/limits and estimate impact on node count and cost

06 Cost Guardrails in CI/CD (IaC Review, Cost Estimation, Policy Checks)

- Why cost must be part of deployment governance
- IaC review workflow and approval gates
- Cost estimation concepts (before apply, environment-specific)
- Policy checks (allowed instance types, region controls, budget thresholds)
- **Hands-on:** Lab: Add a “cost guardrail” step into CI/CD with review and approval logic

07 Dashboards and Reporting for Engineering Teams

- What engineering teams need to see (unit cost, trends, anomalies)
- Dashboards by team/service/environment
- Weekly/monthly reporting workflow and accountability metrics
- Tracking optimization outcomes (savings, regressions, ROI)
- **Hands-on:** Workshop: Design a FinOps dashboard layout and reporting template for engineering leaders

08 Governance (Budgets, Alerts, Anomaly Detection, Accountability)

- Budgets by team/service and environment
- Alerting thresholds and escalation workflow
- Anomaly detection concepts (unexpected spikes, misconfigurations)
- Accountability model (owners, review cadence, FinOps rituals)
- **Hands-on:** Activity: Build a governance checklist for budgets, alerts, and ownership mapping

09 Practical FinOps Implementation Plan for DevOps Organizations

- 30/60/90-day FinOps rollout plan
- Quick wins vs long-term cost engineering roadmap
- FinOps operating model (meetings, reporting cadence, KPIs)
- Sustaining improvements (continuous optimization loop)
- **Hands-on:** Capstone Workshop: Create a FinOps implementation plan for a DevOps organization with actions, owners, and timelines