DEVOPS INTERMEDIATE 3 DAYS

DevSecOps Essentials Bootcamp

Build a strong foundation in implementing a practical DevSecOps pipeline, from secure coding to container security and Kubernetes hardening. Learn how to automate security checks across CI/CD using IaC scanning, policy enforcement, and security gates to deliver fast, compliant, and reliable releases.

Why DevSecOps?

Reduce security risk without slowing delivery

Automated checks catch issues early and prevent insecure releases.

Fewer production incidents

Secure containers, safe Kubernetes defaults, and controlled pipelines reduce outages.

Lower cost of fixing vulnerabilities

PR-phase fixes are significantly cheaper than post-release remediation.

Higher customer trust

Strong security hygiene improves reliability and contract readiness.

Better compliance and auditability

Security controls become repeatable through Git history and pipeline logs.

What you'll be able to do

- Understand DevSecOps fundamentals and how it differs from traditional security models.
- Shift security left by integrating checks into developer workflow and CI pipelines.
- Implement secure SDLC practices including SAST, dependency scanning (SCA), and secret scanning.
- Secure Docker images using best practices, minimal base images, and safe tagging strategies.
- Scan container images for vulnerabilities and enforce policy thresholds in CI.
- Apply Kubernetes security fundamentals including namespaces, RBAC basics, and securityContext hardening.
- Use resource requests and limits to improve workload stability and reduce security risk from noisy neighbors.
- Apply IaC security practices through Terraform scanning and remediation.
- Understand supply chain risks and implement controls like trusted registries and SBOM awareness (conceptual).
- Build an end-to-end GitHub Actions DevSecOps pipeline with PR gating and incident simulation.

Who this is for

- DevOps Engineers
- Cloud Engineers
- SRE Teams
- Backend and Full Stack Developers

- Platform Engineering Teams
- Security Engineers working with DevOps pipelines
- Engineering Leads implementing secure delivery workflows

Curriculum

01 DevSecOps Fundamentals and Shift-Left Security

- DevSecOps definition and goals
- Traditional security vs DevSecOps (shift-left vs end-stage security)
- Shared responsibility across Dev, Sec, Ops
- Security as code (automated checks and policies)
- Where to integrate security checks (IDE/pre-commit, PR, CI)
- **Hands-on:** Lab: Configure GitHub branch protection rules and required checks

02 Secure SDLC Scanning (SAST, SCA, Secrets)

- SAST fundamentals and common findings
- Dependency scanning (SCA) and risk-based prioritization
- Secret scanning patterns and prevention
- Fail vs warn strategy by severity
- **Hands-on:** Lab: Add SAST + SCA + secret scanning to CI and block merge on critical issues

03 GitHub Actions DevSecOps Pipeline Foundations

- GitHub Actions workflow structure (triggers, jobs, steps)
- Runner and permissions basics
- Secure secrets usage (GitHub Secrets, environments)
- PR workflow design for gated security checks
- **Hands-on:** Lab: Build a PR workflow that runs security checks and gates merges

04 Findings Triage and Remediation

- Reading and interpreting scan reports (severity, CVE, fix availability)
- Triage workflow (fix, suppress, accept risk with justification)
- Reducing noise with baselines and allowlists
- Creating a remediation playbook and ownership model
- **Hands-on:** Activity: Triage sample findings and create a prioritized remediation plan

05 Docker Image Security and Trusted Registries

- Common container risks (root user, bloated images, outdated base images)
- Secure Dockerfile practices (multi-stage builds, non-root user)
- Minimal base images concepts (slim/alpine/distroless)
- Preventing secret leakage in images
- Why using 'latest' is risky
- **Hands-on:** Activity: Implement tagging standards and enforce them in CI

06 Container Image Scanning and CI Policy Gates

- Why scanning must be automated in pipelines
- Interpreting vulnerability scans (CVE, severity, fix available)
- Publishing scan reports as pipeline artifacts
- Baseline and exception handling for known risks
- Defining policy thresholds (fail on Critical, warn on High)
- **Hands-on:** Lab: Fail CI when Critical/High vulnerabilities exceed threshold

07 Supply Chain Security and SBOM Awareness

- Software supply chain risk overview
- Risks (compromised dependencies, poisoned images, dependency confusion)
- Controls (trusted registries, least-privilege CI permissions)
- SBOM awareness (what it is and why it matters)
- **Hands-on:** Activity: Generate an SBOM (or review one) and document key insights

08 Kubernetes Security: Namespaces, RBAC, and securityContext

- Namespace isolation fundamentals
- RBAC basics (Role/ClusterRole, RoleBinding/ClusterRoleBinding)
- Least privilege principles in Kubernetes
- Common RBAC misconfigurations and impact
- securityContext essentials (runAsNonRoot, allowPrivilegeEscalation=false)
- **Hands-on:** Lab: Harden a deployment YAML using securityContext best practices

09 Resource Controls for Stability and Risk Reduction

- Requests vs limits explained
- Noisy neighbor problem and availability risks
- Resource controls as a security control (DoS risk reduction)
- Practical sizing starter patterns
- **Hands-on:** Lab: Apply requests/limits to workloads and observe scheduling/stability effects

10 IaC Security with Terraform Scanning

- Common Terraform risks (open security groups, public buckets, permissive IAM)
- Terraform scanning concepts and how CI blocks risky merges
- Remediation workflow (fix module defaults, tighten policies)
- Adding guardrails to prevent regressions
- **Hands-on:** Lab: Scan Terraform code in CI and remediate critical misconfigurations

11 Capstone: End-to-End GitHub Actions DevSecOps Pipeline

- Pipeline stages design (PR checks, scans, build, policy gate)
- PR gating rules (required checks, protected branches)
- Integrating SAST, SCA, secret scanning, image scanning, IaC scanning
- Incident simulation scenarios (secret leak, vulnerable dependency, vulnerable image)
- **Hands-on:** Capstone Lab: Build the complete GitHub Actions DevSecOps pipeline with PR gating and simulate incidents to validate blocking and reporting