DevSecOps Course Content

Chapter 1: Introduction to DevOps

- What is DevOps?
- DevOps Building Blocks – People, Process, Technology
- DevOps Principles (CAMS – Culture, Automation, Measurement, Sharing)
- Benefits of DevOps – Speed, Reliability, Availability, Scalability, Automation, Cost & Visibility
- Common challenges in adopting DevOps

Chapter 2: CI/CD Fundamentals

- What is Continuous Integration and Continuous Deployment?
- From CI → CD → Continuous Delivery
- Continuous Delivery vs Continuous Deployment
- General Workflow of a CI/CD Pipeline
- Blue/Green Deployment Strategy
- Achieving Full Automation

Chapter 3: Designing and Implementing CI/CD Pipelines

- Designing a CI/CD pipeline for web applications
- Hands-on: Building a CI Pipeline using GitLab CI/Jenkins/Travis and GitLab/GitHub/Bitbucket
- Hands-on: Creating a complete CI/CD pipeline

Chapter 4: DevOps Case Studies

- DevOps practices at Facebook, Amazon, and Google
- Lessons from large-scale DevOps adoption

Chapter 5: Tools of the Trade

- GitLab, GitHub, Bitbucket
- Docker basics for DevOps
- CI/CD Tools: Jenkins, GitLab CI, Travis, Bitbucket Pipelines
- Security Testing Tool: OWASP ZAP
- Configuration Management: Ansible
- Compliance & Testing: Inspec

Chapter 6: Secure SDLC in DevOps

- What is Secure SDLC?
- Secure SDLC Activities & Security Gates
- Security across SDLC Phases:
- Requirements (Security Requirements)
- Design (Threat Modeling)
- Implementation (Static Analysis, Secure by Default)
- Testing (Dynamic Analysis)
- Deployment (OS & Application Hardening)
- Maintenance (Security Monitoring & Compliance)

Chapter 7: DevSecOps and Maturity Models

- Introduction to DevSecOps Maturity Model (DSOMM)
- 4 Axes of DSOMM
- Maturity Levels (1 ⟶ 4) with tasks and challenges
- Best Practices for Level 1
- Considerations for Level 2
- Challenges in Level 3
- Vision for Level 4
- Embedding Security in CI/CD pipelines
- DevSecOps challenges with Pentesting & Vulnerability Assessment
- Hands-on: Secure CI/CD pipeline with automated findings management

CREDO SYSTEMZ

Chapter 8: Software Composition Analysis (SCA)

- What is Software Component Analysis?
- Challenges in SCA
- Evaluating SCA solutions (Free & Commercial)
- Embedding tools: OWASP Dependency Checker, Safety, RetireJS, NPM Audit, Snyk
- Demos & Labs:
- OWASP Dependency Checker (Java)
- RetireJS & NPM Audit (JavaScript)
- Safety/Pip (Python)

Chapter 9: Static Application Security Testing (SAST)

- What is SAST?
- Challenges in Static Analysis
- Embedding SAST tools (FindBugs, etc.)
- Secrets Scanning & Prevention
- Writing custom checks for secrets leakage detection

Chapter 10: Dynamic Application Security Testing (DAST)

- What is DAST?
- Challenges in DAST (Session Management, AJAX Crawling)
- Embedding DAST tools: ZAP, Burp Suite
- SSL & Server Misconfiguration Testing
- Creating Baseline Scans for DAST

Chapter 11: Infrastructure as Code (IaC) and Security

- What is Infrastructure as Code & its benefits?
- Platform + Infrastructure Definition + Configuration Management
- Introduction to Ansible
- Benefits of Ansible in IaC
- Push vs Pull based configuration management
- Ansible modules, tasks, roles & playbooks
- Tools & Services for IaC

CREDO SYSTEMZ

Chapter 12: Compliance & Vulnerability Management

- Compliance as Code: Approaches at DevOps scale
- Managing compliance with Inspec/OpenSCAP
- Approaches to vulnerability management
- Hands-On Lab: Using Defect Dojo for vulnerability management