

Foundation DevOps and Security Pipeline

Duration: 8 Days (8Hours per day)

Hands-On Format: This hands-on class is approximately 80/20 lab to lecture ratio, combining engaging lecture, demos, group activities and discussions with comprehensive machine-based practical programming labs and project work.

Prerequisite:

- Basic Linux Knowledge
- Azure free/trial account required

Day 1 and Day 2

Module-1 DevOps Foundation

Exploring DevOps

- Defining DevOps
- Why Does DevOps Matter?

Core DevOps Principles

- The Three Ways
- The First Way
- The Theory of Constraints
- The Second Way
- The Third Way
- Chaos Engineering
- Learning Organizations

Key DevOps Practices

- Continuous Delivery
- Site Reliability & Resilience Engineering
- DevSecOps
- ChatOps
- Kanban

Business and Technology Frameworks

- Agile
- ITSM
- Lean
- Safety Culture
- Learning Organizations
- Sociocracy/Holacracy
- Continuous Funding

Culture, Behaviours & Operating Models



- Defining Culture
- Behavioural Models
- Organizational maturity models
- Target Operating Models

Automation & Architecting DevOps Toolchains

- CI/CD
- Cloud
- Microservices/Containers
- DevOps Toolchain

Measurement, Metrics, and Reporting

- The Importance of Metrics
- Technical Metrics
- Business Metrics
- Measuring & Reporting Metrics

Sharing, Shadowing and Evolving

- Collaborative Platforms
- Immersive, Experiential Learning
- DevOps Leadership
- Evolving Change

Module-2 DevOps Tools (Day 3 & Day 4)

Module 2.1 – GitHub

Introduction to Version Control System Lab: Basic Git Commands Lab: Git Init, Git Add, Git Commit Lab: Working with Remote Repositories Lab: Git Pull and Push Lab: Git Tags

Module 2.2 – Docker & Kubernetes - Basics

Introduction to Docker Lab: Installing Docker Lab: Pulling Image Lab: Running Image from Downloaded Image Lab: Managing Containers – Creating, Deleting, Stopping and Starting Lab: Create Docker Image using Docker Commit Lab: Pushing Image to Docker Hub Basics of Kubernetes Lab: Creating K8s Cluster Lab: Basic Kubernetes Resources – Pod, Service, Labels



Module 2.3 – Jenkins

What is Jenkins Lab: Installing Jenkins Lab: Managing Plugins in Jenkins Lab: Creating Basic Jenkins Job using Freestyle project Lab: Creating Jenkins Pipeline Understanding Jenkinsfile

Module-3 DevOps & DevSecOps Pipeline (Day 5,6,7)

Module 3.1 – Introduction

DevOps vs DevSecOps Security Aspects

Module 3.2 – DevOps Pipeline

Git Repository VM Configuration Lab: Create Azure VM Jenkins Introduction Lab: Jenkins Plugin Installation Jenkins Pipeline – Checking Versions Understanding the Usecase Lab: Running Microservices on Local Machine Maven Basics Lab: Jenkins GitHub Integration and Maven Lab: Build Unit Tests Basic Lab: Unit Test and JaCoCo

Module 3.3 – DevSecOps Pipeline

Git Hooks and Talisman Introduction Lab: Talisman Demo Mutation Tests – PIT Basics Lab: Mutation Tests - PIT Lab: Demo SonarQube introduction SonarQube – Quality Gate SonarQube Authentication Clarification **Vulnerability Basics Dependency Check Basics** Lab: Dependency Check Lab: Demo Refactoring Jenkins **Trivy Basics** Lab: Trivy Image Scan **OPA Conftest Basics OPA Conftest – Docker Kubernetes Security Concepts** Lab: Demo – OPA Conftest Kubernetes **Kubernetes Deployment Rollout Kubesec Basics**



Lab: Kubesec – Demo Fixing Script and ReadOnlyRootFileSystem Trivy – Kubernetes Integration Tests DAST Basics OWASP ZAP Basics Lab: OWASP ZAP – Jenkins Scan Lab: OWASP ZAP – Fixing Issue Lab: OWASP ZAP – Ignore Test Cases

Module 3.4 – Kubernetes Operations and Security

CIS Benchmarking and Kube-bench Lab: Kube Bench Demo Pod-Pod Communication – Need for mTLS Istio Basics Lab: Istio Installation Lab: Istio Injecting SideCar Container Lab: Promoting App to Prod and Visualize using Kiali Istio mTLS Basics Lab: Istio mTLS Demo Lab: Istio Ingress Gateway and VirtualService **Kubernetes Monitoring Basics Prometheus Basics Prometheus Grafana** Falco Basics Lab: Falco Installation - CLI Falco UI – HELM Falco Slack Notifications Lab: KubeScan Demo Integration Tests – Prod

Module-4 Burp Suite & ReactJS- Basics (Day - 8)

Module 4.1 – Security (Free/Trial Version Only)

Introduction to Burp Suite Overview of Burp Suite and its purpose in web application security testing. Key features and editions (Community, Professional, and Enterprise). Setting Up Burp Suite Installation process for Burp Suite. Configuring browser proxy settings to work with Burp Suite. Core Tools in Burp Suite Intruder: Automated payload injection. Repeater: Manually modifying and resending requests. Scanner: Vulnerability scanning capabilities. Proxy: Intercepting and modifying HTTP requests and responses. Performing a Simple Web Application Test Capturing traffic with Burp Suite Proxy. Analyzing HTTP requests and responses. Overview of the ZAP Proxy feature and its role in intercepting web traffic. How it helps identify vulnerabilities by monitoring and modifying HTTP/HTTPS requests and responses. Introduction to Veracode Overview of Veracode as a cloud-based application security platform. Key features: Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software Composition Analysis (SCA), and Manual Penetration Testing. Setting Up Veracode



Creating an account and accessing the Veracode platform. Installing the Veracode CLI for integration with development environments. Configuring Veracode for your first application scan.

Module 4.2 – React JS Overview

React JS vs Other UI Frameworks Installation of Node JS and React library Virtual DOM/ Virtual Memory Component-Based Architecture Lab:: Hello World in React JS