**DEVSECOPS**

**Half-Day 1: Introduction to DevSecOps and Threat Modelling**

* **Overview of DevSecOps**:
  + Definition, Principles, and Benefits
  + Importance of Security in DevOps Culture
* **Understanding the DevSecOps Lifecycle**:
  + Key Phases and Responsibilities
* **Requirements Gathering and Use Case Development**:
  + Gathering and Defining Security Requirements
* **Threat Modelling (ASVS)**:
  + Application Security Verification Standard (ASVS)
  + Identifying Abuse Cases
  + Conducting Risk Analysis
  + Understanding STRIDE Model
* **Hands-On Exercise**:
  + Introduction to a Sample Application
  + Develop Threat Models for a Sample Application
  + Perform Risk Analysis

**Half-Day 2: Security Requirements**

* **SCM (Source Code Management)**:
  + Understanding SCM and Branching Strategies
  + Protected Branches, Rules, and Policies
  + Implementing Approvals and Auto-Merge
* **Git Secrets Management**:
  + Protecting Secrets in Codebases
  + Implementation of Git-secrets
* **Hands-On Exercise**:
  + Implement SCM Security

**Half-Day 3: Secure Coding Practices**

* **Code Review (SAST)**:
  + Understanding White Box Testing
  + Tools and Secure Coding Guidelines
  + Static Application Security Testing (SAST)
  + **SonarQube Installation**:
    - Creating Profiles and Quality Gates
  + Implementing Code Scanning in Visual Studio
  + Integrating Code Scanning in Jenkins
* **Hands-On Exercise**:
  + Implement Code Scanning or SAST in a CI Pipeline

**Half-Day 4: Dependency and CI Server Security**

* **Dependency Scanning or Software Composition Analysis (SCA)**:
  + Dependency Scanning and Code Analysis
  + Understanding NVD (National Vulnerability Database)
  + OWASP Dependency-Check Implementation
  + Security Unit Tests
* **Docker**:
  + Understanding Docker Architecture
  + Creating Containers, Networks, and Volumes
  + Creating Images Using Dockerfile
* **Hands-On Exercise**:
  + Set Up CI Pipeline with Security Scanning Tools

**Half-Day 5: CI/CD Integration with Security**

* **Docker Security**:
  + Best Practices for Securing Docker Containers
  + Analyzing Dockerfile and Image Security
  + Using Tools like Trivy and Hadolint for Dockerfile and Image Security
* **Integrating Security in CI/CD Pipelines**:
  + Continuous Integration Tools (Jenkins, GitLab)
  + Integrating Secure Code Review Tools
* **Hands-On Exercise**:
  + Complete CI/CD Pipeline with SAST, SCA, and Docker Security Checks

**Half-Day 6: Vulnerability Management and Continuous Testing**

* **Dynamic Application Security Testing (DAST)**:
  + Understanding Black Box Testing
  + OWASP ZAP (DAST Tool) Implementation
* **Vulnerability Management Lifecycle**:
  + Identification, Prioritization, and Remediation
  + Continuous Vulnerability Assessment
* **Infrastructure as Code (IaC)**:
  + Role of IaC in DevSecOps
* **Using Terraform for IaC**:
  + Basics of Terraform and Security in IaC
  + Creating Resources Using Terraform
  + Integrating Terraform with Security Policies
* **Hands-On Exercise**:
  + Perform Vulnerability Assessment in a CI/CD Pipeline
  + Automate Infrastructure with Terraform

**Half-Day 7: Configuration Management with Ansible and IAST**

**Introduction to Ansible:**

* **Configuration Management and Automation**
  + Overview of how Ansible helps in managing configurations across environments
* **Writing Playbooks for Secure Deployments**
  + Hands-on with writing Ansible playbooks to enforce security policies
  + Automating common security tasks and compliance checks
* **Implementing Security Policies with Ansible:**
  + Integrating security configurations as code
  + Compliance as Code with Ansible for security audits

**Introduction to IAST (Interactive Application Security Testing):**

* **Understanding IAST:**
  + How IAST works to identify vulnerabilities in running applications
  + Comparison with other testing approaches like SAST and DAST
  + Integrating IAST in a DevSecOps pipeline
* **IAST Demo and Hands-on:**
  + Setting up an IAST tool (e.g., Contrast Security, Veracode IAST) in a sample application
  + Running an IAST scan to detect vulnerabilities during runtime
  + Analyzing IAST reports and remediating issues

**Comparison: SAST, DAST, and IAST**

* **Key Differences and Use Cases:**
  + Static vs. Dynamic vs. Interactive testing
  + Pros and cons of each tool in the context of DevSecOps
  + Combining SAST, DAST, and IAST for comprehensive security coverage

**Hands-On Exercise:**

* **Automate Security Configurations with Ansible**
  + Automate security hardening for servers and applications using Ansible playbooks
* **IAST Setup and Testing**
  + Implement and test IAST on a sample application

**Half-Day 8: Orchestration Tool (Kubernetes)**

* **Introduction to Kubernetes**:
  + Installing Kubernetes
  + Understanding Management Pods
  + Creating Pods, Replica Sets, and Deployments
  + Creating Services
* **Implementing Security Policies in Kubernetes**:
  + Network Policies, Pod Security, and RBAC (Role-Based Access Control)
* **Hands-On Exercise**:
  + Creating an Application and Exposing It Using Kubernetes Services

**Full-Day 9: Final Project and Advanced Topics**

**GitOps:**

* **Understanding GitOps:**
  + How GitOps enables continuous deployment with a focus on security
  + Overview of ArgoCD and its role in GitOps
* **Integrating ArgoCD with Kubernetes and Git:**
  + Deploying applications securely using GitOps workflows
  + Setting up security policies in Kubernetes via GitOps

**Advanced Security Practices:**

* **Orchestration and Risk Management in DevSecOps:**
  + Risk management strategies for DevSecOps pipelines
  + Implementing security automation for governance and compliance
* **Penetration Testing and Feedback Loops:**
  + Continuous feedback on security tests and refining defenses
  + Integrating penetration testing results into the security pipeline

**Leveraging Jenkins Library to Integrate Security Tools:**

* **Introduction to Jenkins Shared Libraries:**
  + How to use shared libraries in Jenkins to standardize security testing stages
  + Leveraging Jenkins libraries to integrate tools like SAST, DAST, IAST, and dependency checks
* **Mandating Security Stages in CI/CD Pipelines:**
  + Implementing mandatory security checks in Jenkins and Azure DevOps pipelines
  + Setting up security gates that block non-compliant code from progressing through the pipeline

**Final Project:**

* **Comprehensive DevSecOps Pipeline Implementation:**
  + Teams implement a fully secured CI/CD pipeline with all tools and methodologies learned
  + Use of Ansible, Kubernetes, Jenkins, IAST, and GitOps in an end-to-end pipeline
* **Group Presentation on Security Strategy and Implementation:**
  + Present a security strategy, discuss methods, and show practical implementation
  + Include detailed documentation, guides, and reference links for all security tools and methods used

**Q&A, Feedback, and Next Steps in the DevSecOps Journey:**

* **Discussion on Real-World Use Cases and Practical Applications:**
  + Case studies from various industries implementing DevSecOps
  + Tips for scaling DevSecOps in large organizations