

# Implementing DevOps Solutions and Practices using Cisco Platforms (DEVOPS)

**Course Duration: 40 Hours**

**Course code: C-DEVOPS**

## 1. Course Overview

The Implementing DevOps Solutions and Practices Using Cisco Platforms course teaches you how to automate application deployment, enable automated configuration, enhance management and improve scalability of cloud microservices and infrastructure processes on Cisco® platforms. Learn to integrate Docker and Kubernetes to create advanced capabilities and flexibility in application deployment.

## 2. What you'll learn?

**After completing this course you should be able to:**

- Describe the DevOps philosophy and practices, and how they apply to real-life challenges
- Explain container-based architectures and available tooling provided by Docker
- Describe application packaging into containers and start building secure container images
- Utilize container networking and deploy a three-tier network application
- Explain the concepts of configuration item (CI) pipelines and what tooling is available
- Implement a basic pipeline with Gitlab CI that builds and deploys applications
- Implement automated build testing and validation
- Describe DevOps principles applied to infrastructure
- Implement on-demand test environments and explain how to integrate them with an existing pipeline

- Implement tooling for metric and log collection, analysis, and alerting
- Describe the benefits of application health monitoring, telemetry, and chaos engineering in the context of improving the stability and reliability of the ecosystem
- Describe how to implement secure DevOps workflows by safely handling sensitive data and validating applications
- Explain design and operational concepts related to using a mix of public and private cloud deployments
- Describe modern application design and microservices architectures
- Describe the building blocks of Kubernetes and how to use its APIs to deploy an application
- Explain advanced Kubernetes deployment patterns and implement an automated pipeline
- Explain how monitoring, logging, and visibility concepts apply to Kubernetes

### 3. Target Audience

Network and software engineers interested in automation and programmability.

### 4. Pre-Requisites

**Attendees should meet the following prerequisites:**

- Basic programming language concepts and familiarity with Python
- Basic understanding of compute virtualization
- Ability to use Linux, text-driven interfaces, and CLI tools, such as Secure Shell (SSH), bash, grep, ip, vim/nano, curl, ping, traceroute, and telnet
- Foundational understanding of Linux-based OS architecture and system utilities

- CCNA® level core networking knowledge
- Foundational understanding of DevOps concepts
- Awareness and familiarity with continuous integration, continuous deployment, and continuous delivery (CI/CD) concepts

## 5. Course content

### 1- Introducing the DevOps Model

- DevOps Philosophy
- DevOps Practices

### 2- Introducing Containers

- Container-Based Architectures
- Linux Containers
- Docker Overview
- Docker Commands

### 3- Packaging an Application Using Docker

- Dockerfiles
- Golden Images
- Safe Processing Practices

### 4- Deploying a Multitier Application

- Linux Networking
- Docker Networking
- Docker Compose

### 5- Introducing CI/CD

- Continuous Integration
- CI Tools
- DevOps Pipelines

### 6- Building the DevOps Flow

- GitLab Overview
- GitLab CI Overview

- Continuous Delivery with GitLab

## **7- Validating the Application Build Process**

- Automated Testing in the CI Flow

## **8- Building an Improved Deployment Flow**

- Post deployment Validation
- Release Deployment Strategies

## **9- Extending DevOps Practices to the Entire Infrastructure**

- Introduction to NetDevOps
- Infrastructure as Code

## **10- Implementing On-Demand Test Environments at the Infrastructure Level**

- Configuration Management Tools
- Terraform Overview
- Ansible Overview
- Ansible Inventory File
- Use the Cisco IOS Core Configuration Module
- Jinja2 and Ansible Templates
- Basic Jinja2 with YAML
- Configuration Templating with Ansible

## **11- Monitoring in NetDevOps**

- Introduction to Monitoring, Metrics and Logs
- Introduction to Elasticsearch, Beats and Kibana
- Introduction to Prometheus and Instrumenting Python Code for Observability

## **12- Engineering for Visibility and Stability**

- Application Health and Performance
- AppDynamics Overview
- Chaos Engineering Principles

### **13- Securing DevOps Workflows**

- DevSecOps Overview
- Application Security in the CI/CD Pipeline
- Infrastructure Security in the CI/CD Pipeline

### **14- Exploring Multicloud Strategies**

- Application Deployment to Multiple Environments
- Public Cloud Terminology Primer
- Tracking and Projecting Public Cloud Costs
- High Availability and Disaster Recovery Design Considerations
- IaC for Repeatable Public Cloud Consumption
- Cloud Services Strategy Comparison

### **15- Examining Application and Deployment Architectures**

- The Twelve-Factor Application
- Microservices Architectures

### **16- Describing Kubernetes**

- Kubernetes Concepts: Nodes, Pods and Clusters
- Kubernetes Concepts: Storage
- Kubernetes Concepts: Networking
- Kubernetes Concepts: Security
- Kubernetes API Overview

### **17- Integrating Multiple Data Center Deployments with Kubernetes**

- Kubernetes Deployment Patterns
- Kubernetes Failure Scenarios
- Kubernetes Load-Balancing Techniques
- Kubernetes Namespaces
- Kubernetes Deployment via CI/CD Pipelines

### **18- Monitoring and Logging in Kubernetes**

- Kubernetes Resource Metrics Pipeline

- Kubernetes Full Metrics Pipeline and Logging

### **Labs:**

- Interact with GitLab Continuous Integration
- Explore Docker Command-Line Tools
- Package and Run a WebApp Container
- Build and Deploy Multiple Containers to Create a Three-Tier Application
- Explore Docker Networking
- Build and Deploy an Application Using Docker Compose
- Implement a Pipeline in Gitlab CI
- Automate the Deployment of an Application
- Validate the Application Build Process
- Validate the Deployment and Fix the Infrastructure
- Build a YAML IaC Specification for the Test Environment
- Manage On-Demand Test Environments with Terraform
- Build Ansible Playbooks to Manage Infrastructure
- Integrate the Testing Environment in the CI/CD Pipeline
- Implement Pre-Deployment Health Checks
- Set Up Logging for the Application Servers and Visualize with Kibana
- Create System Dashboard Focused on Metrics
- Use Alerts Through Kibana
- Instrument Application Monitoring
- Use Alerts and Thresholds to Notify Webhook Listener and Webex Teams Room
- Secure Infrastructure in the CI/CD Pipeline
- Explore Kubernetes Setup and Deploy an Application
- Explore and Modify a Kubernetes CI/CD Pipeline
- Kubernetes Monitoring and Metrics - ELK