
Implementing DevOps Solutions and Practices Using Cisco Platforms

DURATION: 5 DAYS

COURSE CODE: DEVOPS

FORMAT: LIVE/VIRTUAL

COURSE DESCRIPTION

The Implementing DevOps Solutions and Practices Using Cisco Platforms (DEVOPS) v1.0 course teaches you how to automate application deployment, enable automated configuration, enhance management, and improve scalability of cloud micro-services and infrastructure processes on Cisco® platforms. You will also learn how to integrate Docker and Kubernetes to create advanced capabilities and flexibility in application deployment. This course prepares you for the 300-910 Implementing DevOps Solutions and Practices Using Cisco Platforms (DEVOPS) certification exam.

This course will help you:

- Gain the high-demand knowledge and skills to implement automation, streamline container orchestration, and enhance scalability
- Learn the skills to maximize the lightweight design of containers to scale more quickly and allow more responsiveness to website traffic load
- Prepare for the 300-910 DEVOPS exam

The 300-910 DEVOPS exam certifies your knowledge and skills with DevOps. Key DevOps practices include deployment automation that enables configuration, management, and scalability of cloud micro-services and infrastructure processes on Cisco platforms.

After you pass 300-910 DEVOPS, you earn the Cisco Certified DevNet Specialist - DevOps certification, and you satisfy the concentration exam requirement for the CCNP® Collaboration and Cisco Certified DevNet Professional certifications.

WHO SHOULD ATTEND

Account manager
Consulting systems engineer
Network administrator
Network engineer
Network manager
Sales engineer
Systems engineer
Technical solutions architect
Wireless design engineer
Wireless engineer

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
Hands-on experience with Git

LEARNING OBJECTIVES

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 micro-services 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

COURSE OUTLINE

1. **Introducing the DevOps Model**
2. **Introducing Containers**
3. **Packaging an Application Using Docker**
4. **Deploying a Multitier Application**
5. **Introducing CI/CD**
6. **Building the DevOps Flow**
7. **Validating the Application Build Process**
8. **Building an Improved Deployment Flow**
9. **Extending DevOps Practices to the Entire Infrastructure**
10. **Implementing On-Demand Test Environments at the Infrastructure Level**
11. **Monitoring in NetDevOps**
12. **Engineering for Visibility and Stability**
13. **Securing DevOps Workflows**
14. **Exploring Multicloud Strategies**
15. **Examining Application and Deployment Architectures**
16. **Describing Kubernetes**
17. **Integrating Multiple Data Center Deployments with Kubernetes**
18. **Monitoring and Logging in Kubernetes**

DISCOVERY LABS

- 1: **Interact with GitLab Continuous Integration (CI)**
- 2: **Explore Docker Command-Line Tools**
- 3: **Package and Run a WebApp Container**
- 4: **Build and Deploy Multiple Containers to Create a Three-Tier Application**
- 5: **Explore Docker Networking**
- 6: **Build and Deploy an Application Using Docker Compose**
- 7: **Implement a Pipeline in Gitlab CI**
- 8: **Automate the Deployment of an Application**
- 9: **Validate the Application Build Process**
- 10: **Validate the Deployment and Fix the Infrastructure**
- 11: **Build a Yaml Ain't Markup Language (YAML) Infrastructure as Code (IaC) Specification for the Test Environment**
- 12: **Manage On-Demand Test Environments with Terraform**
- 13: **Build Ansible Playbooks to Manage Infrastructure**
- 14: **Integrate the Testing Environment in the CI/CD Pipeline**
- 15: **Implement Pre-deployment Health Checks**
- 16: **Set Up Logging for the Application Servers and Visualize with Kibana**
- 17: **Create System Dashboard Focused on Metrics**
- 18: **Use Alerts Through Kibana**
- 19: **Instrument Application Monitoring**
- 20: **Use Alerts and Thresholds to Notify Webhook Listener and Cisco Webex® Teams™ Rooms**
- 21: **Secure Infrastructure in the CI/CD Pipeline**
- 22: **Explore Kubernetes Setup and Deploy an Application**
- 23: **Explore and Modify a Kubernetes CI/CD Pipeline**
- 24: **Kubernetes Monitoring and Metrics—Elasticsearch, Logstash, and Kibana (ELK)**