Master in DevOps Engineering

STRUCTURE

www.cromacampus.com | helpdesk@cromacampus.com | +91-120-4155255

Master in DevOps Engineering Curriculum

"DevOps: DevOps is a set of Practices that Combines Software Development and IT Operations. It aims to shorten the Systems Development Life Cycle and provide Continuous Delivery with High Software Quality"

About Croma Campus:

Croma Campus Training & Development Private Limited is an education platform since 2010 providing rigorous industry-relevant programs designed and delivered in collaboration with world-class faculty and industry.

- Hands-On Live Projects
- Simulation Test Papers
- Industry Cases Studies
- 61,640+ Satisfied Learners
- 140+ Training Courses
- 100% Certification Passing Rate
- Live Instructor Classroom / Online Training
- 100% Placement Assistance

Croma Campus Training Program Deliverables:

- **Session Recordings** - Original Class Room Voice & Video Recording
- **Training Material** - Soft Copy Handbooks
- **Assignments** | Multiple Hands-on Exercises
- **Test Papers** - We provide **Practice Test** as part of our course to help you prepare for the actual certification exam.
- **Live Case Studies**
- **Live Projects** - Hands-on exercises and Project work. You will work on real time industry-oriented projects and assignments for each module to practice.
- **Key focus on Hands-on exercises and Project work**. You will work on real time industry-oriented projects.
- Faculty with more than **10+ Years of Experience** in the Industry.
- **Technical Resume Designing & Job Assistance:** With more than 100+ Clients across the Globe and we help learners to get a good job in their respective field. We also help learners with resume preparation.
- **Interview Q&A**
- **About Croma Campus Training Certificate:** Croma Campus will provide you with an industry-recognized (Certified by **ISO 9001:2015** & **E-Cell IIT Jodhpur**) course completion certificate which has lifelong validity.
- **How I Unlock my Croma Campus Certificate:** Attend Complete Batch & Submit at least One Completed Project.

This course will provide you the in-depth knowledge of various DevOps tools including **Git, Maven, Jenkins, Nexus Repository, SonarQube, Docker, Ansible, Puppet, Kubernetes and Nagios**. This training is completely hands-on oriented and designed in a way that will help you in becoming a certified practitioner by providing you an intensified training for the best practices about Continuous Development, Configuration Management, including Continuous Integration and Continuous Deployment and finally Continuous Monitoring of the software throughout its development life cycle.

- **Git:** For Version Control for Tracking Changes in the Code Files
- **Maven:** For Software Packaging
- **Jenkins:** For Continuous Integration and Continuous Deployment
- **Nexus Repository:** For Software Component Management
- **SonarQube:** For Code Quality and Security
- **Docker:** For Container Image which is a Lightweight, Executable Package of Software which includes everything needed to run the image (e.g., code, libraries, etc.)
- **Puppet:** Open-source Software Configuration Management Tool
- **Nagios:** Application Monitoring Tool
- **Ansible:** Ansible is an Open-Source Software Provisioning, Configuration Management, and Application-Deployment Tool Enabling Infrastructure as Code.
- **Kubernetes:** Kubernetes is an Open-Source Container-Orchestration System for Automating Computer Application Deployment, Scaling, and Management.

Program Modules:

- **DevOps Certification Training**
- **Terraform Certification Training**
- **Docker Certification Training**
- **Kubernetes Certification Training**
- **Ansible Certification Training**
- **Jenkins Certification Training**
- **Microservices Certification Training**

Master in DevOps Engineering Course Content

Course 1. DevOps Certification Training Curriculum

Module 1: Infrastructure Setup

In this module, you will learn what Cloud Computing is and what are the different models of Cloud Computing along with the key differentiators of different models. We will also introduce you to virtual world of AWS along with AWS key vocabulary, services and concepts.

- EC2 Walkthroug
- Installation of DevOps Tools in the Cloud
    - Git
    - Docker
    - Selenium
    - Jenkins
    - Puppet
    - Ansible
    - Kubernetes

Module 2: Overview of DevOps: In this module you will be introduced to DevOps environment.

- Why DevOps?
- What is DevOps?
- DevOps Market Trends
- DevOps Engineer Skills
- DevOps Delivery Pipeline
- DevOps Ecosystem

Module 3: Introduction to DevOps on Cloud:

Learn about various Cloud Services and service providers; also get the brief idea of how to implement DevOps using AWS.

- Why Cloud?
- Introduction to Cloud Computing
- Why DevOps on Cloud?
- Introduction to AWS
- Various AWS Services
- DevOps using AWS

Module 4: Git, Jenkins & Maven (Version Control, Continuous Integration & Build)

In this module, you will learn about the different actions performed through Git and will be introduced to Jenkins and Maven.

- Version Control with Git
    - What is Version Control
    - What is Git
    - Why Git for your Organization
    - Install Git
    - Common Commands in Git
    - Working with Remote Repositories
- Git, Jenkins & Maven Integration
    - Branching and Merging in Git
    - Git Workflows
    - Git Cheat Sheet
    - What is CI
    - Why CI is Required
    - Introduction to Jenkins (With Architecture)
    - Introduction to Maven
    - Branching and Merging, Stashing, Rebasing, Reverting and Resetting
    - Build and Automation of Test using Jenkins and Maven
- Jenkins (Continuous Integration)
    - Jenkins Management
    - Adding a Slave Node to Jenkins
    - Building Delivery Pipeline
    - Pipeline as a Code
    - Implementation of Jenkins
    - Build the Pipeline of Jobs using Jenkins
    - Create a Pipeline Script to Deploy an Application over the Tomcat Server

Module 5: AWS (Amazon Web Services)

In this module, you will learn about the introduction to compute offering from AWS called EC2. We will cover different instance types and Amazon AMIs. A demo on launching an AWS EC2 instance, connect with an instance and hosting a website on AWS EC2 instance. We will also cover EBS storage Architecture (AWS persistent storage) and the concepts of AMI and snapshots.

- Introduction to Cloud Computing & AWS
- Elastic Compute and Storage Volumes
- Load Balancing, Autoscaling and DNS
- Virtual Private Cloud
- Storage - Simple Storage Service (S3)
- Databases and In-Memory DataStores
- Management and Application Services

- Access Management and Monitoring Services
- Automation and Configuration management
- AWS Migration
- AWS Architect Interview Questions

Module 6: Docker (Continuous Deployment)

This module introduces Docker to Readers, the Core Concepts and Technology behind Docker. Learn in detail about Container and various Operations Performed on it.

- Shipping Transportation Challenges
- Introducing Docker
- Understanding Images and Containers
- Running Hello World in Docker
- Introduction to Container
- Container Life Cycle
- Sharing and Copying
- Base Image & Docker File
- Working with Containers
- Publishing Image on Docker Hub
- Create and Implement Docker Images and Containers

Module 7: Puppet (Continuous Deployment)

Configuration Management with Puppet: In this module, you will learn to Install and Configure Puppet. Additionally, understand the Master-Agent Architecture in Puppet.

- Introduction to Puppet
- Puppet Installation
- Puppet Configuration
- Puppet Master and Agent Setup
- Puppet Module
- Node Classification
- Puppet Environment
- Puppet Classes
- Automation & Reporting
- Install and Configure Puppet
- Configure and Implement Servers using Puppet

Module 8: Ansible (Configuration Management)

Configuration Management with Ansible: In In this module, you will learn to Install Ansible and Configure Ansible Roles. You will also learn to write Playbooks and finally Execute Ad-Commands using Ansible.

- Introduction to Ansible
- Ansible Installation
- Configuring Ansible Roles
- Write Playbooks
- Executing Adhoc Command & Installing Ansible
- Configuring Ansible Role & Write Playbooks
- Execute Adhoc Commands

Module 9: Kubernetes (Containerization using Kubernetes)

Containerization using Kubernetes: In this module, you will learn the basics of Kubernetes and its Integration with Docker.

- Revisiting Kubernetes Cluster Architecture
- Spinning up a Kubernetes Cluster on Ubuntu VMs
- Exploring your Cluster
- Understanding YAML
- Creating a Deployment in Kubernetes using YAML
- Creating a Service in Kubernetes
- Installing Kubernetes Dashboard
- Deploying an App using Dashboard
- Using Rolling Updates in Kubernetes
- Containers and Container Orchestration
- Introduction to Kubernetes
- Setting up the Kubernetes Cluster
- Accessing your application through service
- Deploying an app through Kubernetes Dashboard
- Rolling updates in Kubernetes

Module 10: Nagios (Continuous Monitoring with Nagios)

Continuous Monitoring with Nagios: In Objective: Learn how to Continuously Monitor your Tasks using Various Plugins and Implementing Nagios Commands.

- Introduction to Continuous Monitoring
- Introduction to Nagios
- Installing Nagios
- Nagios Plugins(NRPE) and Objects

- Nagios Commands and Notification
- Installing Nagios
- Monitoring of different servers using Nagios

Course 2. Terraform Training Curriculum

Module 1 - Getting Started & Setting Up Labs

- Choosing a right Infrastructure as Code Tool
- Installing Terraform - Windows Users
- Installing Terraform – Mac OS and Linux Users
- Choosing Right IDE for Terraform IAC development
- Setting up AWS account

Module 2 - Deploying Infrastructure with Terraform

- Creating first EC2 instance with Terraform
- Understanding Resources & Providers
- Destroying Infrastructure with Terraform
- Understanding Terraform State files
- Understanding Desired & Current States
- Challenges with the current state on computed values
- Terraform Provider Versioning
- Types of Terraform Providers

Module 3 - Read, Generate, Modify Configurations

- Understanding Attributes and Output Values in Terraform
- Referencing Cross-Account Resource Attributes
- Terraform Variables
- Approaches for Variable Assignment
- Data Types for Variables
- Fetching Data from Maps and List in Variable
- Count and Count Index
- Conditional Expressions
- Local Values
- Terraform Functions
- Data Sources
- Debugging in Terraform
- Terraform Format
- Validating Terraform
- Configuration Files
- Load Order & Semantics
- Dynamic Blocks

- Tainting Resources
- Splat Expressions
- Terraform Graph
- Saving Terraform Plan to File

Module 4 - Terraform Provisioners

- Understanding Provisioners in Terraform
- Types of Provisioners
- Implementing remote-exec provisioners
- Implementing local-exec provisioners

Module 5 - Terraform Modules & Workspaces

- Understanding DRY principle
- Implementing EC2 module with Terraform
- Variables and Terraform Modules Terraform Registry
- Terraform Workspace
- Implementing Terraform Workspace

Module 6 - Remote State Management

- Integrating with GIT for team management
- Security Challenges in Commiting TFState to GIT
- Remote State Management with Terraform
- Implementing S3 Backend
- Challenges with State File locking
- Integrating DynamoDB with S3 for state locking
- Terraform State Management
- Importing Existing Resources with Terraform Import

Module 7 - Security Primer

- Handling Access & Secret Keys the Right Way in Providers
- Terraform Provider Use Case - Resources in Multiple Regions
- Handling Multiple AWS Profiles with Terraform Providers
- Terraform & Assume Role with AWS STS
- Sensitive Parameter

Module 8: Placement Guide

- Tips to clear an Interview
- Common Interview questions and answers
- Terraform Interview Questions and Answers
- Resume Building Guide
- Attempt for the related Global Certification Exam
- Earn Credentials and Start applying for Jobs

Course 3. Docker Training Curriculum

Module 1: Containerization Overview

- Containerization
- History of Containers
- Namespaces and C Groups
- Containers vs Virtual Machines
- Types of Containers
- Introduction to Docker
- Docker Architecture
- Container Lifecycle
- Docker CE vs Docker EE

Module 2: The Docker Engine

- Docker Engine
    - Setting up Docker Engine
    - Upgrading Docker Engine
- Setting up logging drivers in Docker
- Configuring Logging Drivers
- Docker Terminology
- Port Binding
- Detached vs Foreground Mode
- Starting Containers in different modes
- Docker CLI Commands
- Docker Exec Commands
- Restart Policy in Docker
- Removing Containers

Module 3: Image Management & Registry

- Docker-file
- Write a Docker-file to create an Image
- Docker-file Instructions
- Docker Registry
- Configuring Local Registry
- Build Context
- Docker Image Tags
- Setting up Docker Hub
- Removing Images from the Registry

Module 4: Storage in Docker

- Docker Storage
- Types of Persistent Storage
- Volumes
- Bind Mounts

- TMPFs Mount
- Storage Drivers
- Device Mapper
- Docker Clean Up

Module 5: Orchestration in Docker

- Docker Compose
- Deploy a Multi-container Application using Compose
- Docker Swarm
- Running Docker in Swarm mode
- Docker Service
- Deploying a Service in Swarm
- Service Placement
- Scale Services
- Rolling Update and Rollback
- Docker Stack

Module 6: Networking and Security

- Docker Networking
- Network Drivers
- Bridge Network
- Overlay Network
- Host and Macvlan
- Docker Security
- Docker Content Trust
- Securing the Docker Daemon
- Create and use a User-defined Bridge Network
- Create and use an Overlay Network
- Use Host and Macvlan Network
- Configure Docker to use External DNS
- Signing images using DCT

Module 7: Docker EE and Monitoring

- Docker Enterprise
- Universal Control Plane (UCP)
- UCP Architecture
- Access Control in UCP
- Docker Trusted Registry (DTR)
- Monitoring using Prometheus

Module 8: Docker with Kubernetes

- Kubernetes Core Concepts
- Setup Kubernetes cluster using GKE
- Kubernetes Common Commands

- Pods - Deploy a Pod
- Deployments - Use a Deployment for pod management
- Labels, Selectors, and Annotations
- Services - Deploy different Services
- Persistent Volumes and Persistent Volume Claims
- Use Persistent Storage in Kubernetes
- Storage Classes
- Use Storage Classes

Module 9: Docker with Microservices

- What are Microservices?
- Understanding Microservices and Docket together
- Using Docker for Microservices
- Advantages of using Docker for Microservices
- Design Microservices Architecture with Docker Containers
- Extending the Architecture of a Microservices-Based App with Docker

Module 10: Placement Guide

- Tips to clear an Interview
- Common Interview questions and answers
- Docker Interview Questions and Answers
- Resume Building Guide
- Career roadmap and certifications
- Attempt for related Global Certification Exam
- Start applying for Jobs

Course 4. Kubernetes Training Curriculum

Module 1: Kubernetes Core Concepts and Networking

- Kubernetes Fundamentals
  - Kubernetes Core Concepts
  - Kubectl common commands
  - Understanding Pods
  - Configure network on cluster nodes
  - Pod Networking Concepts
  - Setting up a cluster - Kubernetes Certificates
- Practical Assignment:
  - Perform basic Kubectl commands
  - Deploy pods and use INIT containers to pre-set an environment
  - Configure Kubernetes network using Calico
  - Use certificates to authenticate resources

Module 2: Kubernetes Services and Scheduling

- Services and Controllers
- Service Networking

- Deploy different kinds of services
- Deploy and configure network Load Balancer
- Primitives necessary for self-healing apps
- Effects of resource limiting on pod scheduling
- Configure Kubernetes Scheduler
- Running multiple Schedulers

Module 3: Kubernetes Controllers

- Replica Set and Replication Controller
- Deploy different Replication Controllers
- About Daemon Sets
- Use Daemon Sets on nodes
- Deployments
- Manage pod updates using Deployments
- Rolling updates and Rollbacks
- Scaling applications and Ingress
- Use HPA for dynamic work-load management
- Use Ingress controller and rules to manage network traffic

Module 4: Persistent Storage in Kubernetes

- Persistence Storage Overview
    - Persistent Volume and Persistent Volume Claim
    - Access modes for volumes
    - Primitives for Persistent Volume Claim
    - Secrets and Config Maps in your pods
    - Storage classes
    - Headless services
    - Stateful Sets
- Lab Work:
    - Deploy Persistent Volume and Persistent Volume Claim
    - Use Secrets and Config Maps in your applications
    - Use Storage Class for dynamic storage allocation
    - Use stateful applications for sticky identities for pods
    - Deploy a highly available replicated MariaDB cluster

Module 5: Securing Clusters

- Basic Concepts:
    - Authentication
    - Authorization
    - Kubernetes security primitives
    - Configure Network Policies
    - Security Contexts
- Lab Work:
    - Create and use Roles and Role Bindings
    - Define custom Egress and Ingress policies
    - Use probes and configure a restart policy for pods
    - Define privilege and access control using security contexts

Module 6: Logging & Monitoring Clusters

- Monitor cluster using Prometheus
- Visualize logs using EFK stack
- Deploy jobs to run tasks to completion
- Manage etcd cluster
- Use Helm Charts

Module 7: Troubleshooting Clusters

- Troubleshooting application failures
- Troubleshooting cluster failures

Module 8: Placement Guide

- Tips to clear an Interview
- Common Interview questions and answers
- Kubernetes Interview Questions and Answers
- Resume Building Guide
- Career roadmap and certifications
- Attempt for related Global Certification Exam
- Start applying for Jobs

Course 5. Ansible Training Curriculum

Module 1: Introduction to Ansible

- Evolution of Infrastructure
- Overview of Infrastructure as a Code
- What is Configuration Management
- Ansible Overview

Module 2: Ansible Architecture & Installation

- Ansible Architecture and its working
- Ansible in DevOps
- Installation and Configuration
- Working with Command Line Tools

Module 3: The Playbook Grammar

- Playbook YAML definition
- Playbook terms
- Playbook tasks
- Writing Ansible Playbooks
    - Hosts and Users
    - Variables
    - Tasks
    - Handlers
    - Jinja2 Templates

Module 4: Ansible Modules

- Overview of Modules
- Types of Modules
- Core Modules
- Extras Modules
- Return Values
- Adhoc Commands
- Case Study

Module 5: Ansible Roles

- Overview of Roles
- Role Directory Structure
- Using Roles
- Working with Ansible Galaxy

Module 6: Ansible Tower

- Installing Ansible Tower
- Features of Ansible Tower
- Managing Jobs
- Manage and Track Inventory
- Remote Command Execution
- Case Study

Course 6. Jenkins Certification Training

Module 1: Introduction To CI/CD

- Lesson Introduction
- Traditional Software Development
- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Building the continuous deployment process
- Automated Deployment Pipeline
- CI/CD Tool Selection

Module 2: Getting Started with Jenkins

- Lesson Introduction
- Setting Up Git
- Assisted Practice: Git Set-up
- Setting Up Jenkins
- Assisted Practice: Jenkins Set-up
- Maven Set-up
- Assisted Practice: Maven Set-up

- Exploring Jenkins
- Assisted Practice: Build a Maven Project
- Building a Maven Project with Jenkins

Module 3: Build Jobs and Configurations

- Lesson Introduction
- Understanding Jenkins Build Jobs
- Freestyle Build Jobs
- Assisted Practice: Freestyle Job
- Build Triggers
- Assisted Practice: Scheduled Builds
- Assisted Practice: Polling SCM
- Build Steps
- Jenkins Environment Variables
- Post-build Actions
- Assisted Practice: Post-build Actions
- Using Jenkins with Other Languages
- Assisted Practice: Building Projects with Gradle
- Parameterized Build Jobs
- Assisted Practice: Parameterized Builds
- Assisted Practice: Building From Tags
- Assisted Practice: Remote Triggering Parameterized Builds
- Enabling Security in Jenkins
- Assisted Practice: Enabling Security
- Build Pipelines and Promotions
- Triggering Parameterized Builds

Module 4: Configuring Build Pipelines

- Lesson Introduction
- Introducing Pipelines
- Understanding Pipeline Structure
- Defining a Pipeline
- Declarative Pipeline Syntax
- Scripted Pipeline Syntax
- Assisted Practice: Building Pipelines In Jenkins
- Building a Pipeline
- Continuous Integration Pipeline
- Assisted Practice: Building Continuous Integration Pipelines in Jenkins File
- Building Pipelines from Jenkinsfile
- Assisted Practice: Building Continuous Integration Pipelines in Jenkinsfile
- Snippet Generator
- Unassisted Practice: Snippet Generator
- Global Variable Reference

- Declarative Directive Generator
- Unassisted Practice: Restarting Pipelines
- Multistage Pipeline

Module 5: Automated Testing in Jenkins

- Jenkins as an Automation Testing tool
- Assisted Practice: CI with Junit in Jenkins
- Code Coverage using Jacoco
- Assisted Practice: Code Coverage with Jacoco
- Code Coverage using Clover
- Assisted Practice: Code Coverage with Clover
- Acceptance and Performance Tests in Jenkins
- Assisted Practice: Integrate JMeter with Jenkins
- Testing Code Coverage

Module 6: Code Quality Improvement Using Jenkins

- Code Quality and Jenkins
- Internals of Jenkins Jobs
- Assisted Practice: Find Bugs Integration with Jenkins
- Code Complexity
- Assisted Practice: Coverage Complexity with Jenkins
- Open Tasks & Sonar Qube
- Assisted Practice: Sonar Qube with Jenkins
- Static Code Analysis

Module 7: Automated Deployment and Continuous Delivery

- Introduction to Automated Deployment and Continuous Delivery
- Building the Continuous Delivery Process
- Implementing Automated and Continuous Deployment
- Assisted Practice: Deploying a Python Application
- Assisted Practice: Tomcat and Jenkins
- Assisted Practice: PHP and Jenkins
- Deploying Maven App to Tomcat Server

Module 8: Distributed System in Jenkins

- Introduction to Distributed Architecture
- Assisted Practice: Create Multiple Slave Nodes
- Deep Diving Jenkins Master/Slave Architecture
- Assisted Practice: Assigning Jobs to Specific Slave Nodes
- Distributed Builds

Course 7. Microservices Certification Training

Module 1: Evolution of Microservices

- Monolithic Architecture
- Distributed Architecture
- Service oriented Architecture
- Microservice and API Ecosystem
- Microservices in nutshell
- Point of considerations
- SOA vs. Microservice
- Microservice & API

Module 2: Microservices Architecture

- REST Architecture principles
- Microservice Characteristics
- Inter-Process Communications
- Microservice Transaction Management

Module 3: Microservices Design

- Domain Driven Design
- Big Mud Ball to Sweet Gems
- Untangling the Ball of MUD
- Kill the MUD Ball growth
- Repackaging/Refactoring
- Decouple the User interface and Backend Business Logic
- MUD Ball to Services
- Microservice Design Patterns
- Microservice Architecture Decisions

Module 4: Microservices Security

- Why is Security important?
- Microservice Security Principles
- Microservice Security techniques
- Access Tokens
- OAuth 2.0
- How to secure a Microservice using OAuth 2.0?

Module 5: Microservices Testing

- Testing scenarios and strategy
- Test at Different Levels
- Testing Best Practice for Microservices
- Testing methodology
- How to test Microservices?

Module 6: Microservices Reference Architecture

- Reference Architecture
- Microservice Enabler
- Microservices @ Netflix
- Reading properties in various ways
- Implementing config server

- Setting up Discovery Server
- Setting up Discovery Client
- Overview of Actuator Endpoints
- API Gateway and Dynamic Routing
- Declarative Rest Client
- Hystrix Fault Tolerance
- Distributed Caching
- Distributed Sessions
- Need for Event Driven Systems
- Building Event Driven Systems
- Implementing Distributed Tracing
- Understanding Metrics
- Monitoring Microservices
- Spring Boot Admin

Module 7: Placement Guide

- Tips to clear an Interview
- Common Interview questions and answers
- Microservices Interview Questions and Answers
- Resume Building Guide
- Career roadmap and certifications
- Attempt for related Global Certification Exam
- Start applying for Jobs