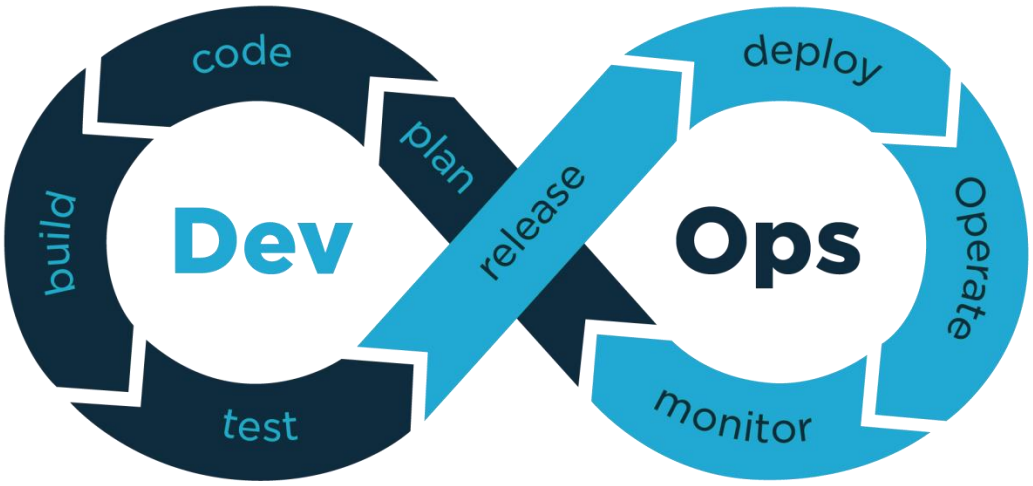
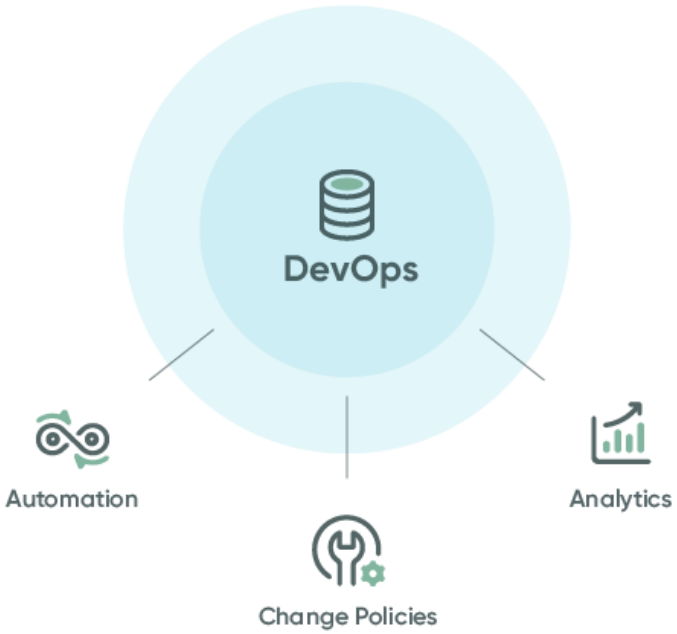


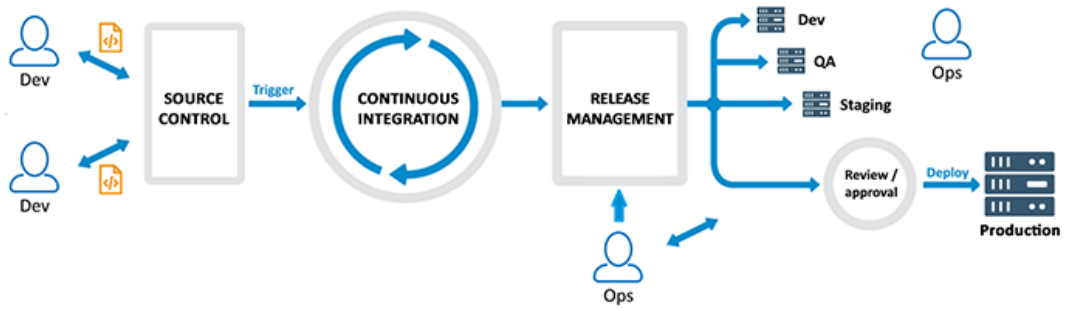
DevOps Certification Training Curriculum



G-21, Sector-03, Noida-201301
Contact#: 0120-4155255
helpdesk@cromacampus.com

DevOps

(Git, Maven, Jenkins, Nexus Repository, SonarQube, Docker, Ansible, Puppet, Kubernetes, and Nagios)



- **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 DevOps Training Program:** Our DevOps training is designed keeping in mind the latest trend in the world of technologies.
  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.)

G-21, Sector-03, Noida-201301
Contact#: 0120-4155255
helpdesk@cromacampus.com

- **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.

DevOps Fundamentals

1. **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

2. **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

3. **Linux Introduction.**
4. **Basic Scripting.**
5. **Virtualization.**
    - Introduction
    - History of Virtualization
    - Types of Virtualization
    - Hypervisor & Benefits of Hypervisor
    - Deep dive into Virtual machine
    - Compute Calculation Networking Storage
    - Installing VirtualBox on Windows Installing OS on the newly created VM
6. **Vagrant**
    - Introduction
    - Installing Vagranton Windows Vagrant Cloud
    - VagrantFile
    - Creating VM'susing Vagrant Deep dive into Vagrant file
    - Vagrant commands
    - Networking
    - Provisioning Synced Folders
    - Configuring Multiple machines
7. **Cloud Computing**
    - Introduction
    - Types of Services
    - Advantages and Disadvantages of CloudComputing
8. **Amazon WebServices**
    - Regions and Availability zones AWS services
    - Identity Access Management AmazonEC2
    - Amazon EBS
    - Amazon VPC
    - AWS Autoscaling
    - Elastic LoadBalancer
    - AmazonS3 AmazonRDS
    - Lambda
    - Cloud watch
    - Amazon DB Services (Theory)
9. **GoogleCloud Platform Basics**
    - Compute Storage
    - Kubernetes Engine

Git, Jenkins & Maven

(Version Control, Continuous Integration & Build)

G-21, Sector-03, Noida-201301
Contact#: 0120-4155255
helpdesk@cromacampus.com

1. **Version Control with Git:** In this module you will be introduced to DevOps environment.
    - What is Version Control
    - What is Git
    - Why Git for your Organization
    - Install Git
    - Common Commands in Git
    - Working with Remote Repositories
    - GIT Installation, Version Control, Working with Remote Repository

2. **Git, Jenkins & Maven Integration:** In this module, you will learn about the different actions performed through Git and will be introduced to Jenkins and Maven.
    - 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

Jenkins
(Continuous Integration)

G-21, Sector-03, Noida-201301
Contact#: 0120-4155255
helpdesk@cromacampus.com

1. **Continuous Integration using Jenkins:** In this module, you will know how to perform Continuous Integration using Jenkins by Building and Automating Test Cases using Maven.
    - 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

Nexus Repository
(Software Component Management)

1. **Software Component Management Using Nexus**: In this module, you will learn Nexus fundamentals and how to manage various software components using Nexus Repository.
    - Nexus Repository Manager
    - What is Nexus?
    - What is Repository Manager?
    - Software component management in Nexus
    - Installation of Nexus
    - Configuration of Nexus
    - Creating a repository
    - P2 Nexus Plugins
    - Nexus Unzip Plugin
    - Nexus Repository Lifecycle
    - Job roles and responsibilities of a repository manager
    - Difference between repository and repository manager
    - Know about SDLC basics and models
    - Repo for SDLC Management

SonarQube
(Code Quality and Security)

1. **SonarQube Basics to Advanced training**: In this module, you will get the basic idea of the tool, its components, what are the coding rules, Code quality standards, and security concepts.
    - **Overview**
        - Why SonarQube?
        - Installation
        - Configurations
        - Browsing/analyzing source code
    - **Components**
        - Plugin Library
        - Local and Branch Analysis
        - Web Services API
    - **Coding rules**
        - Coding Rule Basics
        - Upgrading rules
    - **Code Quality**
        - How Code Quality?
        - Issue Tracker
        - Quality Gate
    - **Security Concepts**
        - Security-related Rules
        - Built-in Rule Tags
        - Reports & Notifications
        - User Account

Docker
(Continuous Deployment)

1. **Continuous Deployment: Containerization with Docker:** 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

Puppet

(Continuous Deployment)

1. **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

Ansible

(Configuration Management)

1. **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

Kubernetes

(Containerization using Kubernetes)

1. **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

Nagios

(Continuous Monitoring with Nagios)

1. **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

Project

Project 1 - Workflow of CI/CD with AWS ,GIT,Jenkins

- Deploy HTML website from GitHub to AWS Ec2 Instance.
- Create Build, Deploy, and test the Jenkins Pipeline.
- Run Jenkins Pipeline automation on a developer.
- Deploy code to EC2 instance on a new commit in GitHub and Run automation test as part of the pipeline.

- Use SSH to connect to the AWS Ec2 Instance from windows.

Finally, integrate the automation testing using CI/CD using git, Jenkins, AWS as crucial aspects in DevOPs and Agile Workflows.

Project 2 - Workflow with Docker, Kubernetes, Git, Jenkins, Ansible

- Deploy a Website with Python Django.
- Create Build, Deploy, and Test the Jenkins Pipeline for Dockers.
- Run Jenkins Pipeline automation on a developer.
- Deploy the code onto the Docker instance.
- Run the Configuration management using Ansible.
- Deploy & Run the Application on the Container Platform.