

## DevOps Course Curriculum

### Overview of DevOps

**Goal:** In this module, you will be introduced to the DevOps environment.

**Objectives:** After completing this module, you should be able to

- Understand the benefits of DevOps over other software development processes
- Gain insights into the DevOps environment
- Get an overview of different DevOps Tools
- Get a picture of the working of the DevOps Delivery Pipeline

**Topics:**

- Introduction to DevOps
- Benefits of working in a DevOps environment
- DevOps Lifecycle
- DevOps Stages
- DevOps Delivery Pipeline

### Version Control with Git

**Goal:** In this module, you will gain insights into Source Control Management and learn the functionalities of Git.

**Objectives:** After completing this module, you should be able to

- Understand Version Control
- Perform management of files for small as well as large projects
- Perform various Git commands such as git add, git fetch, git commit, git init, etc.
- Work with remote repositories

**Topics:**

- Version Control
- Git Introduction
- Git Installation
- Commonly used commands in Git
- Working with Remote repository

## Hands On:

- Git Common Commands
- Working with Remote Repository

## Git, Jenkins & Maven Integration

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

**Objectives:** After completing this module, you should be able to:

- Execute branching and merging operations
- Perform various Git commands
- Understand Maven Architecture and dependencies
- Learn about Continuous Integration & its importance
- Understand Jenkins and its features

## Topics:

- Branching and merging in Git
- Merge Conflicts
- Stashing, Rebasing, Reverting and Resetting
- Git Workflows
- Introduction to Maven
- Maven Architecture
- Introduction to Continuous Integration

- Introduction to Jenkins

## Hands On:

- Branching and Merging
- Merge Conflicts
- Stashing, Rebasing, Reverting, and Resetting
- Configuring Maven

## Continuous Integration using Jenkins

**Goal:** In this module, learn how to perform Continuous Integration by building applications with the help of Maven and create deployment pipelines using Jenkins.

**Objectives:** After completing this module, you should be able to

- Managing authorization in Jenkins
- Jenkins notification management
- Master-slave architecture in Jenkins
- Add a slave node to Jenkins master
- Build and deploy codes using Jenkins
- Build pipeline plugin in Jenkins
- Use Declarative pipeline in Jenkins

## Topics:

- Jenkins Architecture
- Plugin Management in Jenkins
- Jenkins Security Management
- Notification in Jenkins
- Jenkins Master-slave architecture
- Jenkins Delivery Pipeline
- Jenkins Declarative pipeline

## Hands On:

- Create pipeline view using DevCompile and QUnitTest
- Adding Slave node in Jenkins
- Build Pipeline project using Groovy script

## Configuration Management Using Ansible

**Goal:** Learn how to manage and configure your infrastructure using Ansible Ad-Hoc commands, Playbooks, and Roles.

**Objectives:** After completing this module, you should be able to

- Utilize Ansible CLI
- Execute Ansible Ad-Hoc Commands for one-off tasks
- Automate host servers using Ansible Playbooks
- Use Variables in Playbooks
- Using Handlers

## Topics:

- Introduction to Configuration Management
- Infrastructure as Code
- Introduction to Ansible
- Ansible Architecture
- Inventory Management
- Ansible Modules
- AD-HOC Commands
- Ansible Playbooks
- Ansible Roles

## Hands On:

- Ad-Hoc Commands

- Running a Simple Playbook
- Using Variables and handlers
- Using Ansible Roles

## Containerization using Docker Part – I

**Goal:** This module introduces learners to the core concepts and technology behind Docker. Learn in detail about containers and various operations performed on them.

**Objectives:** After completing this module, you should be able to

- Understand Containerization
- Learn the evolution of virtualization to containers
- Understand the Docker Architecture
- Perform Various actions using Docker CLI
- Bind container ports to the Machine ports
- Run containers in different modes
- Write and build a Dockerfile to create a Docker Image

### Topics:

- Containerization
- Namespaces
- Docker
- Docker Architecture
- Container Lifecycle
- Docker CLI
- Port Binding
- Detached and Foreground Mode
- Dockerfile
- Dockerfile Instructions
- Docker Image

## Hands On:

- Docker CLI Commands
- Port Binding
- Starting Containers in Different Modes
- Writing a Dockerfile to Create an Image

## Containerization using Docker Part – II

**Goal:** Learn how to use Docker Hub registry, deploy a multi-tier application using Docker Compose, and create a swarm cluster.

**Objectives:** After completing this module, you should be able to

- Use Docker Hub to store custom Images
- Store data in Container Volumes for persistent storage
- Setup Docker Compose
- Deploy a multi-container application using Docker Compose
- Deploy a Swarm Cluster

## Topics:

- Docker Registry
- Container Storage
- Volumes
- Docker Compose
- Docker Swarm

## Hands On:

- Setting up Docker Hub
- Docker Volumes
- Installing Docker Compose
- Installing a Multi-Container Application using Compose

- Running Docker in Swarm Mode

## Orchestration using Kubernetes Part - I

**Goal:** In this module, you will learn about Container Orchestration and Basic of container management using Kubernetes.

**Objectives:** After completing this module, you should be able to

- Understand Container Orchestration
- Learn about Kubernetes Core Concept
- Deploy Pods
- Create Deployments to manage Pods
- Launch DaemonSets for Background applications
- Update and Rollback your Deployments
- Scale your containerized Applications

### Topics:

- Introduction to Container Orchestration
- Kubernetes Core Concepts
- Understanding Pods
- ReplicaSet and Replication Controller
- Deployments
- DaemonSets
- Rolling Updates and Rollbacks
- Scaling Application

### Hands On:

- Kubectl Common Commands
- Deployments
- DaemonSets

- Rolling-update and Rollbacks
- Scaling in Kubernetes

## Orchestration using Kubernetes Part - II

**Goal:** Learn and deploy different service discovery mechanisms, utilize Volumes for persistent storage and deploy StatefulSets for stateful applications.

**Objectives:** After completing this module, you should be able to

- Deploy different Kubernetes Services
- Utilize Volumes to store Persistent Data
- Create Persistent Volume Claims to attach volumes to Pods
- Understand Persistent Volume Claims Primitives
- Use Headless Services in Stateful Sets
- Deploy Helm Charts

### Topics:

- Services
- Persistent Storage in Kubernetes
- Primitives for PersistentVolumeClaims
- Secrets and ConfigMaps
- Headless Services
- StatefulSets
- Helm Charts

### Hands On:

- Deploying Services
- Persistent Volumes and Persistent Volume Claims
- StatefulSets
- ConfigMaps and Secrets

- Helm Charts

## Monitoring using Prometheus and Grafana

**Goal:** In this module, you will learn how to collect, monitor, and visualize data using Prometheus and Grafana.

**Objectives:** After completing this module, you should be able to

- Understand Continuous Monitoring
- Use Prometheus to monitor services
- Create an alerting mechanism using Prometheus
- Deploy Grafana dashboards to visualize data
- Integrate Prometheus and Grafana to monitor a full pipeline

### Topics:

- Introduction to Prometheus and Grafana
- Prometheus and Grafana Setup
- Monitoring using Prometheus
- Dashboard Visualization using Grafana
- Creating a Dashboard to monitor the Pipeline

### Hands On:

- Monitoring Service using Prometheus
- Alerting using Prometheus
- Grafana Dashboards
- Monitoring a Pipeline

## Provisioning using Terraform Part - I

**Goal:** Learn how to provision and manage infrastructure on a Cloud Platform (AWS) using Terraform Configuration Files.

**Objectives:** After completing this module, you should be able to

- Understand Provisioning using Terraform
- Learn the Difference between Terraform vs Ansible
- Understand Terraform Architecture
- Deploy a Terraform Configuration File
- Use Basic Terraform Commands
- Manage Terraform Resources

**Topics:**

- Introduction to Terraform
- Terraform vs Ansible
- Terraform Architecture
- Terraform Configuration
- Terraform Common Commands
- Managing Terraform Resources

**Hands On:**

- Setting Up AWS and Terraform
- Executing a Terraform Configuration
- Managing Terraform Resources
- Referencing Terraform Resources

## Provisioning using Terraform Part - II

**Goal:** Use Terraform State commands to manage the current state of your infrastructure. Deploy a fully usable and working infrastructure using Terraform.

**Objectives:** After completing this module, you should be able to

- Perform Terraform State Commands
- Deploy a Terraform Project on AWS

## Topics:

- Terraform State
- Terraform Project

## Hands On:

- Terraform State Commands
- Terraform Project

## Selenium (Self-Paced)

**Goal:** In this module, you will learn about selenium and how to automate your test cases for testing web elements. You will also get introduced to X-Path, TestNG and integrate Selenium with Jenkins.

**Objectives:** After completing this module, you should be able to

- Learn and install Selenium
- Create Test Cases in Selenium WebDriver
- Utilize X-Path and TestNG to locate elements
- Execute code on several browsers using Selenium suite of tools
- Integrate Selenium with Jenkins

## Topics:

- Introduction to Selenium
- Why Selenium?
- Selenium – Webdriver
- Creating Test Cases in Selenium WebDriver (Waits)
- What and why X-Path
- Handling different controls on Webpage
- Framework in Selenium
- Selenium Integration with Jenkins
- Implementation of Selenium in the Edureka's Project

## Hands-On:

- Installing Selenium
- Creating Test Cases in Selenium WebDriver
- Integrating Selenium with Jenkins

## Nagios (Self-Paced)

**Goal:** Learn how to continuously monitor your tasks using various plugins and implementing Nagios Commands

**Objectives:** After completing this module, you should be able to

- Operate Continuous Monitoring tools
- Use various plugins and objects associated with Nagios
- Implement Nagios commands

## Topics:

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

## Hands-On:

- Installing Nagios
- Monitoring of different servers using Nagios

## DevOps on Cloud (Self-Paced)

**Goal:** Learn about various cloud services and service providers, also get the brief idea of how to implement DevOps using AWS.

**Objectives:** After completing this module, you should be able to

- Understand about cloud and its advantages
- Learn about Various cloud computing services
- Get an idea of how to implement DevOps using AWS

## Topics:

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

## AWS EC2 and IAM (Self-Paced)

**Goal:** Get a brief idea of how Security and EC2 Compute service works in AWS Cloud.

**Objectives:** After completing this module, you should be able to

- Describe AWS Global Infrastructure and its Benefits
- Sign-up an AWS free-tier account
- Work with AWS Management Console and AWS CLI
- Work with IAM Service
- Understand Virtualization
- Work with EC2
- Analyze various EC2 box configurations available

## Topics:

- Virtualization
- Amazon Web Services (AWS)
- Benefits of AWS
- AWS Global Infrastructure

- AWS: IAM
- Components of IAM
- Managing users with IAM
- Amazon Machine Image (AMI)
- Security Groups in AWS
- Virtualization
- Amazon Elastic Compute Cloud (EC2) and Its Benefits
- Networking components associated with EC2
- Instance Store

## **Hands-on:**

- Signing up for a Free Tier Account with AWS
- Creating New User to Log in to AWS Management Console
- Creating Policies for New User to Have All Admin or Limited Privileges
- Different Approaches to connect to an EC2 instance
- Creating a Custom AMI
- Host your Website Inside your EC2 Instance
- To Attach EFS Volume to an EC2 Instance
- Login to AWS Console via MFA

## DevOps Certification Project

### What are the system requirements for this DevOps course?

You don't have to worry about the system requirements as you will be executing your practicals on a Cloud LAB which is a pre-configured environment. This environment already contains all the necessary tools and services required for Edureka's DevOps Training online.

### How will I execute the practicals in this DevOps course?

You will execute all your DevOps Course Assignments/Case Studies in the Cloud LAB environment provided by Edureka. You will be accessing the Cloud LAB via a browser. In case of any doubt, Edureka's Support Team will be available 24\*7 for prompt assistance.

### Which case studies or projects are a part of this DevOps training course?

#### **Project 1:**

AppleBite Co. is using Cloud for one of their products. The project uses modular components, multiple frameworks and want the components to be developed by different teams or by 3rd-party vendors. The company's goal is to deliver the product updates frequently to production with High quality & Reliability. They also want to accelerate software delivery speed, quality and reduce feedback time between developers and testers. As development progressed, they are facing multiple problems, because of various technologies involved in the project. Following are the problems:

- Building Complex builds is difficult
- Incremental builds are difficult to manage, and deploy

To solve these problems, they need to implement Continuous Integration & Continuous Deployment with DevOps using following tools:

- **Git:** For version control for tracking changes in the code files
- **Jenkins:** For continuous integration and continuous deployment

- **Docker:** For deploying containerized applications
- **Ansible:** Configuration management tool

## Project 2:

A retail company Abstergo Corp. has recently setup an online shopping portal(website) to sell their products. Due to fierce competition, the company wants a solution that can reduce the time and effort it needs to enhance its website functionality on a regular basis. They are looking for an automated way to deploy the new code (for new features) to the production website whenever they want. Steps to follow:

- The team of developers working on new features will merge their code to a GitHub repo
- As soon as the code reaches GitHub, using a CI (Continuous Integration) pipeline, setup in Jenkins, an automated build will be triggered.
- The automated build will frequently deploy new features to the production website.
- Every build will prepare a Dockerfile and push docker images to docker-hub.
- Every docker image will be deployed (Continuous Deployment) to a kubernetes-cluster.

## What is CloudLab?

CloudLab is a cloud-based DevOps environment that Edureka offers with the DevOps Training Course where you can execute all the in-class demos and work on real-life DevOps case studies fluently. This will save you from the trouble of installing and maintaining DevOps tools on a virtual machine. You'll be able to access the DevOps Training CloudLab via your browser which requires minimal hardware configuration. In case, you get stuck in any step, our support team is ready to assist 24x7.