

educationnest.com



**Education  
Nest**



# DevOps

6 Months Instructor-led live  
online Certificate Course

# What is DevOps?

The combination of cultural philosophies, practices, and tools increases an organization's ability to deliver applications and services at a high speed.

## So, what does a DevOps Engineer do?

**A DevOps engineer, therefore, has responsibilities that also extend to the deployment of the developed software as per the project requirements.**

There are tools under this model that help engineers independently accomplish tasks that normally involve other teams, thereby increasing a team's efficiency and velocity.

**A DevOps Engineer is the backbone of any organization's technical team.**

According to Forbes, DevOps and Cloud jobs will be amongst the highest-paying jobs in the near future. Consequently, this industry is expected to grow by six times by the year 2024 which opens jobs for those aspiring to succeed in this field.

# DevOps Certification Training Course Objective

Sambodhi Research and Communications anticipates this dynamic opportunity, which is why it brings to you DevOps Certification Training Course, an instructor-led, live, virtual training that will help you acquire DevOps training online, and master various aspects expected from the role.

In this DevOps certification course, **you will master various aspects of software**

**development, operations, continuous integration, continuous delivery, automated build, testing, and deployment.**

Furthermore, you will gain insights into the automation of configuration management, inter-team collaboration, and IT service agility using DevOps tools such as Git, Puppet, Jenkins, SVN, Maven, Docker, Ansible, Nagios, and more.



## Education Nest

### About Education Nest

As a subsidiary of Sambodhi Research and Communications Pvt. Ltd., Education Nest is a global knowledge exchange platform that empowers learners with data-driven decision-making skills. It features online courses

designed by experts to help you expand your skills and engage with a global network of learners through live training sessions, case-based learning, and interactions with the best from the field.




### About Sambodhi

Sambodhi Research has over 16 years of experience and the ability to create high-impact knowledge and accountability in the public domain through data science and evidence-based solutions. As an organization with the ability to field build and nurture outcomes-driven ecosystems, we relentlessly

pursue our vision of empowering individuals with the best learning programs. Our focus is to enable each student to appear as an 'industry-ready' professional, equipped with the tools needed for a successful career in the Data Science industry.

# Key Highlights:

-  Industry-relevant curriculum designed by industry experts.
-  180 hours of instructor-led live virtual training on weekends
-  60 hours of Capstone project and exercises
-  1-year subscription to Education nest's enterprise-grade Learning Management System (LMS)
-  Globally accredited recognition for students
-  Availability of scholarships and EMI payment options for DevOps training
-  Weekend Instructor-Led Live Virtual classes
-  live online classes conducted by industry experts.
-  compact and industry-relevant course curriculum,
-  Live Mentor Support
-  Customized Professional Learning

## What will you learn from this course?

This DevOps Certification Course will help you develop the key skills necessary to become a DevOps expert. Other than training in configuration management, continuous integration, deployment, delivery, and monitoring using

DevOps tools. With this course, you will also learn the fundamentals of Agile and Scrum methodologies and gain knowledge of the two major cloud platform providers—Amazon Web Services (AWS) and Microsoft Azure.



Git



SVN



Docker



Docker Swarm



Kubernetes



Puppet



Ansible



Selenium

**Maven**



## Furthermore, this DevOps training will also help you gain expertise in:

- DevOps methodology
- Implementing software version control
- Containerizing code on production using Docker
- Creating CI/CD pipelines using Jenkins
- Configuration management using Puppet and Ansible
- Automating build and test using Selenium and Maven
- Container orchestration using Kubernetes, and
- Performance tuning and monitoring using Nagios

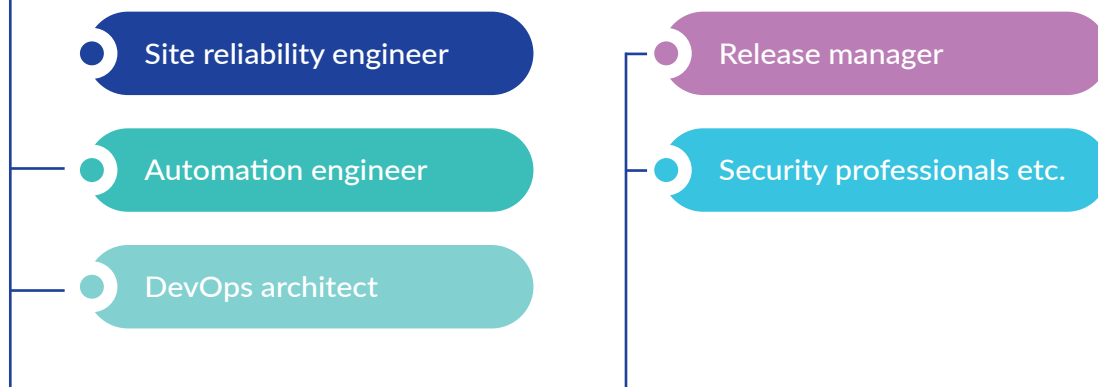
## Who should enroll in this course?

Anyone working in Operations or Development and Engineering teams can opt for a career in DevOps. Students need to have a graduate degree or professional IT diploma. Technical background and a basic understanding of Linux, web development, and Java programming are necessary.

## Individuals working in the following roles will benefit the most from the DevOps Certification course:

- IT Team Leaders
- Software Developers
- Systems Administrators and IT Managers
- Cloud Engineers
- Developers
- IT professionals
- Software testers
- System administrators
- Solutions architects
- Security engineers
- Application developers
- Integration specialists

### ● On successful completion of this certificate course, you will be eligible for roles such as:



# Course Contents

## DevOps Fundamentals

### M1 - Introduction to Software Development and DevOps

- Overview of Software Development Life Cycle (SDLC)
- Benefits of SDLC
- Types of SDLC
- Waterfall Model
- Iterative Model
- Agile Development Model
- Challenges with Traditional Development Models
- How DevOps helps solve various Industry Problems Agile with DevOps
- Agile vs DevOps
- DevOps Market Trends

### M2 - DevOps Lifecycle and Tools

- Source code management
- Continuous integration
- Continuous testing
- Continuous deployment
- Configuration management
- Continuous monitoring
- DevOps tools

### M3 - Introduction to DevOps on Cloud

- Introduction to Cloud
- Computing Infrastructure as Code (iac)
- Overview of Cloud Services for DevOps
- AWS DevOps
- Azure DevOps
- Google Cloud
- Managing basic Services
- Creating Instances on the Cloud

# Source Code Management Using Git

## M1 - Getting Started with Git

- Version control system
- Need for VCs (version control system)
- What is a VCs?
- Types of VCs
- Centralized
- Distributed
- Introduction to git
- What is git?
- Need for git
- Use-cases
- Getting started with git
- Git terminology
- Configuring git
- Command-line options
- Environment variables
- Configuration files
- Using the git config tool
- System level configuration
- Global level configuration
- Local level configuration
- Configuration commands
- List, unset, get, edit, etc.
- Client vs server configuration

## M2 - Working with Git Repositories

- Overview of GitHub
- Git Repositories
- Local and Remote Repositories
- Working With Remote Repositories
- Initializing a Repository
- Cloning a Repository
- Displaying remotes
- Adding a Remote
- Fetching and Pulling from Remotes
- Pushing to Remote
- Inspecting a Remote
- Renaming and Removing a Remote
- Forking a Repository
- Git File Life Cycle
- Recording changes to the Repository
- Checking Status
- Tracking Files
- Staging Files
- Viewing Staged and Unstaged Changes
- Committing Changes
- Checking the Commit History
- Removing Files
- Moving Files
- Undoing Things - Reset, Checkout, Restore

## M3 - Branching and Merging in Git

- Branching in Git
- What is a Branch?
- Benefits of Branching
- Internal management of Branches
- Git Storage Strategy
- Object
- Head
- Origin Master
- Types of Branches
- Switching between branches and different commits
- Deleting Branches
- Two-way merge and Three-way merge
- Merge conflicts
- Resolving a merge conflict
- Merging strategies
- Resolve, recursive, octopus, ours, subtree, fast forward
- Merging and rebasing
- What is merging?
- What is rebasing?
- How does rebasing work?
- Advantages of rebasing
- Merging vs rebasing
- Git tags
- What is a tag?
- Types of tags
- Operations on tags



## M4 - Git Workflows

- M4 - git workflows
- What is a git workflow?
- Advantages of a workflow
- Several types of workflows in git
- Centralized workflow
- Feature branch workflow
- Gitflow workflow
- Forking workflow

## M5 - Using Git in IDE

- Git Plugins with Eclipse
- Install Eclipse IDE
- Install IntelliJ IDE
- Configure the Git plugin with Eclipse
- Work with Git using Eclipse
- Work with Git Workflow on Eclipse IDE
- Explore GitHub UI Options"

## M6 - Git Server Administration

- Git Server setup
- User Management
- Access Control for SSH and HTTP access
- SSH vs HTTPS -differences between the HTTPS and SSH Authentication Strategies
- Using our system's terminal to generate a public and private SSH key pair that will allow us to use SSH authentication to log into our remote GitHub repository
- Using our newly generated GitHub key to set up SSH authentication to ease the process of pushing up to GitHub from our local system
- Gitlab
- Installation
- Administration
- Working using GitLab

# Continuous Integration Using Jenkins

## M1 - Introduction to Continuous Integration

- Introduction to continuous integration (CI)
- Need for CI
- Use-cases
- Applications of Jenkins/Use-cases
- Getting Started with Jenkins
- Jenkins Architecture and terms of Jenkins
- Configuring Jenkins: Configure System, Environment
- Variables, URL
- Creating a Jenkins Job
- Configuring a Jenkins job
- Advantages of CI
- Overview of Jenkins
- Features of Jenkins
- Overview of Jenkins UI

## M2 - Jenkins Plugins

- Introduction to Plugins
- Need for Plugins
- What is a Plugin?
- Adding Plugins to Jenkins
- Commonly used plugins
- List common plugins e.g., Git, Docker, Jira, Maven,
- Slack Integration, Kubernetes, Pipeline, Amazon EC2,
- Performance, Performance Publisher
- Configuring Jenkins to work with
- Git and GitHub
- Maven
- Overview of Maven
- Why Use Maven?
- What is Maven?
- Maven Vs Gradle Vs Ant
- Working on Maven
- Maven Build Life Cycle
- Creating a Jenkins Build and Jenkins workspace
- Configure Jenkins to check for source code changes periodically
- Collaborating with Maven Build Jobs"

## M3 - Build Jobs and Jenkins Security

- Jenkins pipeline
- Jenkins continuous delivery pipeline
- Pipeline concepts
- Pipeline syntax
- Example/(s)
- Build jobs
- Creating a freestyle build job
- Introduction to build triggers and build steps
- Pre-and post-build actions: adding properties and properties files
- Running your new build job
- Parameterized builds
- Distributed builds
- Setting up email notifications
- Securing Jenkins
- Access control
- Disabling security

## M4 - Metrics to Improve Quality

- Looking for foul Code through Code Coverage
- Activating and using of PMD Jenkins plugin
- Activating and using of Find bugs Jenkins plugin
- Verifying HTML Validity
- Reporting with java n CSS
- Jenkins with Gradle script build system
- Jenkins with shell script build system

## M5 - Managing and Monitoring Jenkins

- Managing Jenkins
- Gathering system information
- User management
- Monitoring Jenkins
- Plugins for reporting
- Plugins for code analysis
- Server maintenance
- Backup plugin
- Managing plugins
- Installing, uninstalling, and updating plugins
- Remote testing

## M6 - Automated and Continuous Deployment

- Deployment Overview
- Automated deployment
- Continuous deployment
- Deploying an application to an application server
- Install and configure tomcat
- Deployment of simple java web application using tomcat
- Jenkins builds a pipeline
- Parallel Jenkins build
- Archive-generated artifacts
- Jenkins integrations (GitHub, slack, custom API)
- Scaling Jenkins"

## M7 - Jenkins Pipeline

- Overview of Pipeline as code
- Overview of Pipeline Plugin
- Automated Jenkins Pipeline

# Continuous Integration Using Jenkins

## M1 - Introduction to Continuous Integration

- Introduction to Configuration Management (CM)
- Need for CM
- Use-cases
- What is infrastructure Orchestration?
- Infrastructure as Code
- Overview of Ansible
- Features
- Ansible Vs Others
- Configuring Ansible
- Ansible architecture
- Inventory management
- Build/create ansible inventory files
- Ansible modules
- Ansible roles directory structure
- Create ansible tasks
- Write ansible ad-hoc commands
- System facts

## M2 - Server Provisioning using Ansible CLI and Playbooks

- Ansible CLI
- What is the Ansible playbook?
- Organizing Tasks in Playbook
- Use handlers and Flow Control in Ansible scripts
- Introduction to YAML
- Writing a Playbook
- Using Variables in Playbooks
- Variables at runtime using ansible Facts

## M3 - Ansible Modules and Roles

- What are Ansible modules
- Using modules in Playbook
- Introduction to Ansible roles
- Manage inclusions or tasks and roles into another role
- Using roles in Ansible Playbook.
- Configure Jinja2 templates
- Define, execute, and manage roles

## M4 - Ansible on Cloud

- Launch EC2 Instance on AWS (REHL or CENTOS)
- Install Ansible on AWS.
- Preparing Ansible to work with AWS
- Module support for Docker interaction

## M5 - Automating Continuous Integration

- Project based on Git, Jenkins, and Ansible

## M6 - Infrastructure Automation using Terraform

- Introduction to Terraform
- Why Use Terraform?
- What is Terraform?
- Use-cases
- Terraform Vs Others
- Using Terraform
- Terraform HCL
- Configuration Management using Ansible and Terraform
- Building an Infrastructure
- Updating an Infrastructure
- Destroying an Infrastructure
- Variables
- Software and Server Provisioning
- Attributes
- Remote State
- Modules

# Continuous Deployment Using Docker

## M1 - Introduction to Containerization

- Containerization
- History of Containers
- Namespaces and Cgroups
- Containers vs Virtual Machines
- Types of Containers
- Introduction to Docker
- Docker Architecture
- Container Lifecycle
- Docker CE vs Docker EE
- Configuring Logging Drivers
- Docker Terminology
- Port Binding
- Detached vs Foreground Mode
- Docker CLI
- Docker Exec
- Restart Policy

## M2 - Image Management and Registry

- Docker file
- Docker file Instructions
- Build Context
- Docker Image
- Docker Registry

### M3 - Storage in Docker

- Docker storage
- Types of persistent storage
- Volumes
- Bind mounts
- Tmpfs mount
- Storage drivers
- Device mapper
- Docker clean up

### M4 - Container Orchestration using Docker

- Docker compose
- Docker swarm
- Docker service
- Service placement
- Rolling update and rollback
- Docker stack

### M5 - Networking and Security

- Docker networking
- Network drivers
- Bridge network
- Overlay network
- Host and macvlan
- Docker security
- Docker content trust
- Securing the docker daemon

### M6 - Docker EE

- Docker Enterprise
- Universal control plane (ucp)
- Ucp architecture
- Access control in ucp
- Docker trusted registry (DTR)

## Container Orchestration Using Kubernetes

### M1 - Introduction to Container Orchestration

- Containerization recap
- What is container orchestration?
- Managing containers at scale
- Container orchestration tools
- Advantages of orchestration
- Introduction to Kubernetes
- Kubernetes architecture

### M2 - Kubernetes Architecture and Core Concepts

- Kubernetes core concepts
- Node status and node lease
- Kubectl common commands
- Understanding pods
- Configure network on cluster nodes
- Pod networking concepts
- Setting up a cluster - Kubernetes certificates

### M3 - Kubernetes Services and Scheduling

- Services and Controllers
- Service Networking
- 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

### M4 - Kubernetes Controllers

- Replica set and replication controller
- Daemon sets
- Deployments
- Rolling updates and Rollbacks
- Scaling applications
- Ingress

### M5 - Storage in Kubernetes

- 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

### M6 - Securing and Troubleshooting the Cluster

- Authentication
- Authorization
- Kubernetes security primitives
- Configure Network Policies
- Security Contexts
- ETCD Operations
- Jobs
- Helm Charts
- Troubleshooting application failures
- Troubleshooting cluster failure

### M7 - CI/CD using Jenkins X

- What is Jenkins X?
- Setting up EKS with Jenkins X
- QuickStart projects
- Kubernetes Cluster with Jenkins X

### M8 - CI/CD using Jenkins X Part-2

- Full Pipeline with Jenkins-X P

# Continuous Monitoring and DevOps on AWS

## M1 - Continuous Monitoring using Prometheus

- Introduction to Continuous Monitoring
- Getting started with Prometheus and Grafana
- Prometheus Architecture
- Prometheus Configuration
- Monitoring Servers with Prometheus
- Pushing Metrics
- Client Libraries and Exporters
- Querying
- Service Discovery
- Introduction to alerting
- Setting up an alerting system

## M2 - Prometheus Security and Use Cases

- Prometheus storage
- Prometheus security
- Grafana provisioning
- Monitoring a web application
- Scraping Kubernetes with Prometheus
- Consul integration with Prometheus
- Ec2 auto-discovery

## M3 - Continuous Monitoring using Nagios XI

- Introduction to Nagios XI
- Nagios Architecture
- Setting up and Configuring Nagios XI
- Managing Users, Components, and Plugins
- Dashboards
- Monitoring using Nagios XI
- Views
- Scheduled Reporting
- Advanced Visualization
- Capacity Planning
- Audit Logging

## M4 - Log Server 2.0 and alerting in Nagios

- Log-server 2.0 architecture
- Log-server 2.0 interface
- Alerting on Data
- Setting up Alerting system
- Configuring Inputs, Outputs, and Filters
- Global Settings
- Auto Discovery
- Notifications

## M5 - SDLC automation using AWS

- Important Phases Involved in AWS DevOps
- AWS code commit
- Versioning in a code commit
- Working on AWS code commit
- AWS code pipeline
- Working on AWS code pipeline
- Concepts of code pipeline
- Input and Output Artifacts
- AWS code build
- Deployment Workflow: On A Lambda Platform and
- Working on AWS code build
- AWS code deploys
- Primary Components
- On EC2 Platform
- Application Specification File
- Deployment Types: In-Place Deployment and
- Blue/Green Deployment
- AWS code star
- Working on AWS code star

## M6 - Automating Infrastructure using CloudFormation

- Infrastructure as code
- CloudFormation
- CloudFormation template
- Working on cloud formation
- Advanced CloudFormation Concepts
- Nesting
- Wait for Conditions and Wait for Handlers
- Helper Scripts
- Custom Resources
- Intrinsic Functions & Conditions
- Update Stack: Direct Update, Update Behavior, Change Sets, Stack policy
- CloudFormation Resource Deletion Policy
- CloudFormation Troubleshoots
- CloudFormation Best Practices

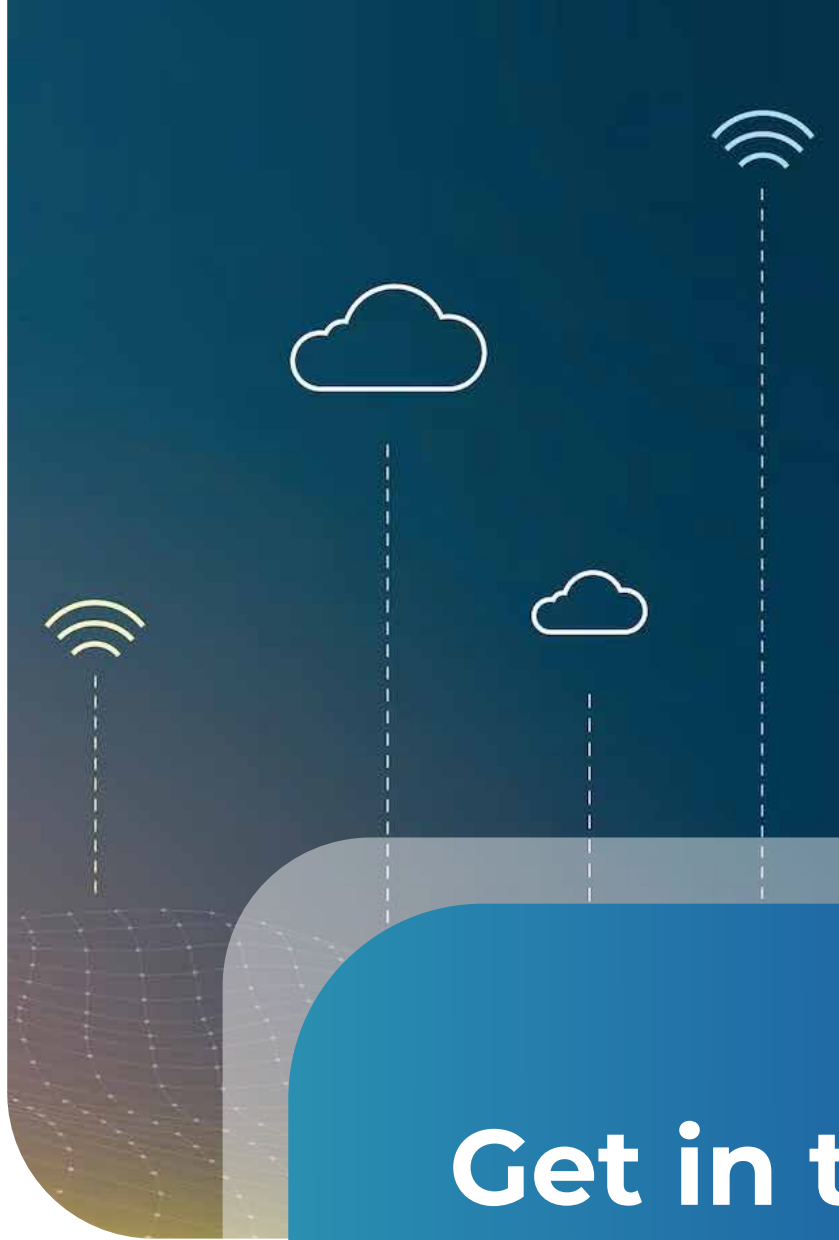
## M7 - Application Deployment using Elastic Beanstalk

- Elastic beanstalk
- Concepts of the elastic beanstalk
- Permissions
- Introduction to containers
- Application environment components
- Container v/s VM
- Docker: docker file, docker image, docker container
- Docker in beanstalk
- Deployment options
- Comparing the deployment methods
- Classification of platform updates
- Configuration files
- Alarms and notification
- Troubleshooting
- Beanstalk v/s CloudFormation

## M8 - Application deployment and Orchestration

- Using ECS, ECR, and EKS
- ECS Introduction
- Need for ECS
- ECS Cluster
- ECS Service with a load balancer
- Deploy an application over the ECS cluster
- Deploy web app with MySQL on EKS Kubernetes Cluster
- Autoscaling of cluster using AWS EKS
- ECR Introduction
- Need for ECR
- Push an image to ECR
- EKS
- Need for EKS
- EKS Cluster
- ECS Task Definition
- ECS Service





# Get in touch with us

**A:** Sambodhi Research and Communications Pvt. Ltd.  
C - 126, Sector 2, Noida - 201301, Uttar Pradesh

**P:** +91 120 4056400-99  
+91 80695 55055

**E:** [trainings@sambodhi.co.in](mailto:trainings@sambodhi.co.in),  
[course@educationnest.com](mailto:course@educationnest.com)

