

Infrastructure as Code (Ansible + Terraform)

Duration: 5 Days (40 Hours)

Hands-On Format: Instructor and participants perform the labs together.

Course Overview

This 5-day hands-on course provides a practical foundation in **Infrastructure as Code (IaC)** using **Ansible and Terraform**.

Participants will learn how to automate infrastructure provisioning, configuration, and deployment using industry-standard DevOps tools.

The course begins with **Git and GitHub fundamentals** for managing and versioning infrastructure code. It then covers **Ansible** for configuration management, system automation, playbooks, templates, roles, and secure automation practices. The **Terraform** section focuses on provisioning cloud infrastructure, variables, state management, remote backends, modules, and reusable configurations. Participants will also learn the fundamentals of **Jenkins CI/CD** and its integration with Ansible and Terraform. The course concludes with a practical **end-to-end IaC workflow** covering GitHub, Jenkins, Terraform, and Ansible.

Prerequisites

- Basic Linux command-line knowledge
- Basic networking concepts
- Basic understanding of cloud or virtualization concepts
- No prior Ansible or Terraform experience required

Git and GitHub Fundamentals

Module 1 – Git and GitHub Fundamentals

- Introduction to Version Control and Git
- Git Architecture and Basic Workflow
- Installing and Configuring Git
- Git Repository and Basic Git Commands
- Working with Commits
- Git Branching and Merging
- Working with Remote Repositories
- GitHub Repository Basics
- **Lab: Install and configure Git**
- **Lab: Create repository, commit changes, and view history**
- **Lab: Create branches and perform merge operations**
- **Lab: Create GitHub repository and push local code**

Automation with Ansible

Module 2 – Ansible Fundamentals

- Infrastructure Automation and Configuration Management
- Introduction to Ansible
- Ansible Architecture
- Control Node and Managed Nodes
- Ansible Installation and Configuration
- Ansible Inventory
- Ansible Configuration File
- Ansible Ad-Hoc Commands
- Ansible Modules and Common Modules
- **Lab: Install and configure Ansible**
- **Lab: Configure Linux managed nodes**
- **Lab: Execute Ansible Ad-Hoc commands**

Module 3 – Ansible Playbooks and Automation

- YAML Fundamentals for Ansible
- Ansible Playbook Structure
- Tasks and Modules
- Variables and Facts
- Conditions
- Loops
- Handlers
- Jinja2 Templates
- Basic Ansible Roles
- Ansible Vault Fundamentals
- **Lab: Create and execute an Ansible Playbook**
- **Lab: Use variables, conditions, and loops**
- **Lab: Configure services using handlers**
- **Lab: Deploy configuration files using Jinja2 templates**
- **Lab: Create and use a basic Ansible Role**

Infrastructure Provisioning with Terraform

Module 4 – Terraform Fundamentals

- Infrastructure as Code Concepts
- Introduction to Terraform
- Terraform Architecture and Workflow
- Terraform vs Ansible

- Terraform Installation and Configuration
- Terraform Providers
- Azure Authentication
- Terraform Configuration Files
- Terraform Resources
- Terraform Workflow: init, validate, plan, apply, destroy
- **Lab: Install and configure Terraform**
- **Lab: Configure Azure Provider and Authentication**
- **Lab: Create an Azure Resource Group**
- **Lab: Provision basic Azure resources using Terraform**

Module 5 – Terraform Configuration, State and Backend

- Terraform Resource Attributes
- Input Variables and Data Types
- Output Values
- Local Values
- Data Sources
- Terraform State
- Desired State vs Current State
- Terraform Backend
- Local vs Remote Backend
- Azure Storage Backend
- State Locking and Concurrent Access
- Provider Version Constraints
- Basic Terraform Debugging
- **Lab: Use Variables and Outputs**
- **Lab: Reference Attributes between Resources**
- **Lab: Use Data Sources**
- **Lab: Inspect and Understand Terraform State**
- **Lab: Configure Azure Storage Backend**
- **Lab: Verify Remote State and State Locking**

Module 6 – Terraform Reusable Infrastructure

- Terraform Expressions
- `count` and `for_each`
- Terraform Modules
- Module Structure
- Local Modules
- Terraform Registry Modules
- Basic Terraform Best Practices
- **Lab: Use `count` and `for_each`**
- **Lab: Create a reusable Terraform Module**
- **Lab: Deploy Infrastructure using a Terraform Module**

CI/CD Fundamentals

Module 7 – Jenkins with Ansible and Terraform

- Introduction to CI/CD
- Jenkins Architecture and Fundamentals
- Jenkins Installation
- Jenkins Jobs and Pipelines
- Jenkins Credentials
- GitHub Integration
- Jenkins Pipeline Fundamentals
- Jenkinsfile
- Executing Ansible from Jenkins
- Executing Terraform from Jenkins
- Infrastructure Deployment Workflow
- **Lab: Install and configure Jenkins**
- **Lab: Connect Jenkins with GitHub**
- **Lab: Create a basic Jenkins Pipeline**
- **Lab: Execute Ansible Playbook from Jenkins**
- **Lab: Execute Terraform from Jenkins**

Module 8 – End-to-End Infrastructure Automation

- IaC Workflow with Git, Terraform and Ansible
- End-to-End Infrastructure Deployment Workflow
- **Lab: GitHub → Jenkins → Terraform → Azure Infrastructure → Ansible Configuration → Application Deployment**