# Terraform Essentials

**Days:** 3

**Prerequisites:** Although not required, students with some experience programming or pre-existing knowledge of cloud architecture will most appreciate the technical nature of this hands-on course.

**Audience:** This course is designed for DevOps Engineers, Software Developers, Technical Managers and Leads, System and Cloud Administrators, and Network Engineers and Developers.

**Description:** As enterprises seek to deploy and maintain increasingly complex cloud infrastructure, there is a necessity to use "Infrastructure as Code" (IaC) tools, like Terraform. An open-source, state management tool developed by HashiCorp, Terraform allows developers to use a common coding interface to work through their various clouds safely and efficiently. Attendees will leave being able to write and understand Terraform code (HCL), have a clear understanding of Terraform's various components and supporting tools, as well as when to reach for Terraform over another IaC tool, such as Ansible.

**Course Objectives:** In this course, you will:

- Writing Terraform HCL code
- Deploying into common clouds such as AWS, Azure, Google Cloud, Docker, Oracle, Kubernetes, and VMWare
- Where Terraform fits in the Enterprise CI/CD model
- Differences between Terraform and Ansible
- Best practices
- Prepare for HashiCorp's Terraform Associate Certification
- AI LLM prompt engineering for Terraform snippets and jumpstarting solutions

**OUTLINE:**

## SOURCE CODE MANAGEMENT
• Lecture: What to Choose?
• Lecture + Lab: SCM Option #1 - GitHub
• Lecture + Lab: SCM Option #2 - GitLab
• Lecture + Lab: gitignore for Terraform

## INTRODUCTION TO TERRAFORM
• Lecture: Terraform Course Map
• Lecture: Introduction to Terraform
• Lecture + Lab: Terraform Install

## TERRAFORM MODULES
• Lecture: Terraform HCL Syntax
• Lecture + Lab: Up and Running with Terraform
• Lecture: Terraform Variables
• Lecture + Lab: Terraform Variables
• Lecture: Terraform Locals
• Lecture + Lab: Output Values
• Lecture: Avoid the :latest Tag
• Challenge: Terraform and Docker

## BEYOND BASICS
• Lecture + Lab: Terraform CLI Workspaces
• Lecture + Lab: Terraform Expressions and Errors
• Lecture + Lab: Resources - Replace vs Taint
• Lecture + Lab: Dynamic Operations with Functions
• Lecture + Lab: Creating a Terraform Module
• Lecture + Lab: Moving State - terraform state mv
• Lecture + Lab: Dynamic Provisioning with tfvars Files
• Lecture + Lab: Data Sources and HTTP Provider
• Lecture: Import Pre-Existing Infrastructure
• Lecture + Lab: Challenge - AWS Import

## LOOPS
• Lecture: for_each
• Lecture + Lab: Looping Constructs - for_each

## PROVISIONING
• Lecture + Lab: local-exec Provisioner
• Lecture + Lab: Creating Delays
• Lecture + Lab: Terraform - templatefile Function

# Terraform Essentials

## TERRAFORM CLOUD
• Lecture + Lab: Terraform Cloud and Terraform Enterprise
• Lecture + Lab: Triggering Cloud Builds via Git Commits

## DYNAMIC BLOCKS
• Lecture + Lab: Dynamic Blocks

## AWS
• Lecture + Lab: Terraform and AWS
• Lecture + Lab: Output Values and AWS
• Lecture + Lab: AWS and Looping with count vs for_each
• Lecture + Lab: Correcting Resource Drift and AWS
• Challenge: Terraform and AWS

## AZURE
• Lecture + Lab: Terraform and Azure

## GOOGLE CLOUD PLATFORM
• Lecture + Lab: Terraform and Google Cloud Platform

## ORACLE
• Lecture + Lab: Terraform and Oracle Cloud Infrastructure

## TERRAFORM AND ENTERPRISE
• Lecture + Lab: Deploy a Go RESTful API Microservice with Terraform
• Lecture: Terraform vs. Ansible
• Lecture + Lab: Terraform and Ansible

## VMWARE
• Lecture: Terraform and VMWare
• Lecture + Lab: Terraform and VMWare

## HELPFUL DEVOPS TOOLS (OPTIONAL)
• Lecture + Lab: Open Policy Agents and Terraform
• Lecture + Lab: GitHub Actions - GitLeaks
• Lecture + Lab: GitHub Actions - Terraform
• Lecture: Terragrunt

## TERRAFORM ASSOCIATE EXAM
• Lecture: Mock Exam