

**Terraform for Azure**

**Course Number:** TRFM-106WA  
**Duration:** 3 days

**Overview**

This Terraform for Azure training course teaches attendees how to use Terraform to deploy and manage resources on Azure. The course starts with the basics of Terraform and then covers IaC (infrastructure as code) and the Terraform workflow, the Azure Terraform provider, HCL (Hashicorp Configuration Language) programming, best practices, and troubleshooting. Students learn how to integrate Terraform with CI/CD pipelines, GitOps, and other Azure and Hashicorp tools.

**Prerequisites**

Attendees must have a basic working knowledge of Linux and Windows-based systems (i.e., Bash or PowerShell) and have a fundamental understanding of Azure.

**Materials**

All Terraform for Azure attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

A modern web browser and an Internet connection free of restrictive firewalls, so that the student can connect by SSH or Remote Desktop (RDP) into AWS virtual machines.

**Objectives**

* Understand IaC and how it is different from traditional scripting
* Initiate a Terraform project and use the Terraform CLI
* Provision and manage Azure resources using Terraform
* Explore Terraform's architecture and how it impacts IaC workflows
* Code in HCL using best practices
* Understand HCL Data Source blocks
* Work with custom modules
* Understand Terraform state management
* Debug Terraform scripts and apply IaC principles
* Implement Terraform within CI/CD pipelines
* Integrate GitOps with Terraform

**Outline**

* Introduction
* Infrastructure as Code Overview
  + Holistic Overview
  + IaC vs Scripting
* Terraform Basics - Getting Started
  + Terraform CLI
  + Terraform Project Structure
  + Terraform Workflow: Plan, Apply, Destroy
  + Terraform HCL Configuration Files
  + Resources - Introduction
  + Providers - Introduction
  + Provisioning Infrastructure using Terraform
  + Using Terraform Provider Documentation
* Getting Started with Terraform and Azure
  + Creating and managing Azure resources using Terraform
  + State files and out-of-band changes - reality check
* Building Infrastructure
  + Terraform Architecture
    - Terraform Development Architecture
    - Terraform State
    - Local State
    - Remote State
    - Intro to Terraform Enterprise and Terraform Cloud
    - Terraform Registry
    - Providers
    - Plugins
    - Modules
    - Terraform CLI vs Provider version pinning
    - CLI Compatibility Promise
    - Provider Backward Compatibility
* Azure Provider Fundamentals
  + Provider Landscape for creating cloud resources
  + Azure Provider Authentication
  + Resource Groups
  + Compute: Azure Virtual Machines, Azure Functions
  + Storage: Azure Blob Storage, Azure Disk Storage
  + Databases: Azure SQL Database, Cosmos DB
  + Network: Virtual Network, Subnets, Route Tables, Internet Gateways
  + Identity: Azure Active Directory Users, Roles, Role Assignments
  + Architecting Infrastructure in Azure with Terraform
  + Terraform vs ARM and Bicep
* Skills Deep Dive
  + HCL Programming Basics
    - Variables and Outputs
    - Input
    - Local
    - Output
    - Input Variables in Detail
    - CLI Parameters
    - TFVARS File
    - Environment Variables
    - Validation Rules
    - HCL Expressions Basics
    - Value Data Types
    - Value References
    - Strings
    - Conditionals
    - For
    - Splat
    - Operators
    - Using Functions in Terraform
    - Numeric
    - String
    - Collections
    - Date and Time
    - Type Conversion
    - Encoding
    - Filesystem
    - Cryptographic and Hashing
    - IP Network
    - Terraform Best Practices - Basics
    - HCL Coding Best Practices
    - Terraform Project Best Practices
  + Resources Deep Dive
    - Deep dive into HCL resource blocks
    - Resource Blocks
    - Resource Behavior
    - Resource Dependencies
    - Implicit
    - Explicit
    - Meta Arguments
    - count
    - depends\_on
    - for\_each
    - provider
    - lifecycle
  + Data Sources
    - Data Source Blocks
    - Working with Azure Data Sources
  + Modules Deep Dive
    - Modules Architectural Overview
    - Writing Custom Modules
    - Consuming Custom Modules
    - Module Meta Arguments
    - Publishing Modules
    - Modules Best Practices
  + Terraform State Deep Dive
    - Terraform State Architecture
    - Terraform Backends
    - Local
    - Remote - Enterprise
    - Remote - Cloud
    - Remote - Azure Storage
    - Managing State
    - State Drift
    - Out of Band Changes
    - Convention over Control
    - Importing Existing Resources into Terraform
    - Terraform CLI Import Command Usage
    - 3rd Party Tools
* Best Practices and Troubleshooting
  + Debugging and Troubleshooting
    - Terraform Log Levels
    - Debugging Terraform Scripts
    - Audit Trails for Troubleshooting
    - Terraform State File Issues
    - Common Terraform Errors and Solutions
  + Terraform Best Practices
    - Infrastructure as Code Principles
    - Mutable vs Immutable
    - Declarative Code
    - Version Control
    - Automation
    - Resource Management Strategies
    - Naming Conventions
    - Modularization
    - Resource Dependencies
    - Lifecycle Management
    - Effective Use of Azure Data Sources
    - Security Best Practices
    - Secrets Management
    - Principle of Least Privileges
    - Use of Service Roles
    - Performance Optimization Tips
    - Parallel Resource Creation
    - Reducing Interdependencies
    - Azure Best Practices
    - IaC Code Management
    - Compatibility and Security
    - Testing and Documentation
    - Resource Management
    - Environment and Automation
* Advanced Concepts
  + Terraform in Organization Context
    - Centralized vs. Distributed Infrastructure Management
    - Hybrid Infrastructure Management
    - DevOps Approach with Terraform
    - Using Managed Terraform Services
    - Team Roles and Responsibilities
    - Governance and Best Practices
    - Case Studies and Real-world Scenarios
  + Last Mile Configuration using Provisioners
    - Configuration vs State
    - Integrating configuration management tools
  + Terraform in CI/CD Pipelines and GitOps
    - Overview of CI/CD
    - Terraform in CI/CD Context
    - GitOps with Terraform
    - Automated Testing of Terraform Code
    - Infrastructure Deployment Automation
    - Case Studies and Examples
    - Best Practices and Tools
  + Integration with Other Tools
    - Azure Key Vault for Secrets Management
    - Azure Monitor for resource monitoring
    - Secrets Management with Vault
    - Service Discovery with Consul
    - Image Creation with Packer
    - Development Environments with Vagrant
    - Application Deployment with Nomad
* Conclusion