

**Terraform for AWS**

**Course Number:** TRFM-100WA
**Duration:** 3 days

**Overview**

Terraform is a cloud infrastructure management tool used to manage AWS deployments. This Terraform for AWS training course demystifies the Terraform workflow and IaC (infrastructure as code), guides learners on using the Terraform CLI, and teaches practical skills for AWS infrastructure provisioning. The course then goes deeper into Terraform architecture, taking a closer look at the details of the Hashicorp AWS provider. In addition, participants refine their HCL (Hashicorp Configuration Language) programming skills, troubleshoot their scripts, incorporate best practices, and integrate Terraform with other 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 AWS 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 the Terraform workflow
* Use the Terraform CLI
* Provision AWS infrastructure
* Perform the steps for provisioning and maintaining diverse AWS resources via Terraform
* Understand Terraform state management
* Code in HCL using best practices
* Debug and troubleshoot a Terraform project
* Integrate Terraform with CI/CD pipelines, GitOps, and other AWS and Hashicorp tools

**Outline**

* Introduction to Terraform
* 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 AWS
	+ Creating and managing AWS resources using Terraform
	+ State files and out-of-band changes - reality check
* 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
* AWS Provider Fundamentals
	+ Provider Landscape for creating cloud resources
	+ AWS Provider Authentication
	+ Creating AWS resources:
	+ Compute: EC2 Instances, Lambda Functions, PaaS Applications
	+ Storage: S3 Buckets, EBS Volumes
	+ Databases: RDS, DynamoDB
	+ Network: VPC, Subnets, Route Tables, Internet Gateways
	+ Identity: IAM Users, Roles, Policies and attachments
	+ Architecting Infrastructure in AWS with Terraform
	+ Terraform vs CloudFormation and CDK
* 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 AWS 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 - AWS S3
	+ Managing State
	+ State Drift
	+ Out of Band Changes
	+ Convention over Control
	+ Importing Existing Resources into Terraform
	+ Terraform CLI Import Command Usage
	+ 3rd Party Tools
* 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 Data Sources
	+ Security Best Practices
	+ Secrets Management
	+ Principle of Least Privileges
	+ Use of Service Roles
	+ Performance Optimization Tips
	+ Parallel Resource Creation
	+ Reducing Interdependencies
	+ AWS Best Practices
	+ IaC Code Management
	+ Compatibility and Security
	+ Testing and Documentation
	+ Resource Management
	+ Environment and Automation
* 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
	+ AWS Secrets Manager for Secrets Management
	+ AWS CloudWatch for resource monitoring
	+ Secrets Management with Vault
	+ Service Discovery with Consul
	+ Image Creation with Packer
	+ Development Environments with Vagrant
	+ Application Deployment with Nomad
* Conclusion