Faysal Aziz

Terraform Automation

Build, Automate and Scale Modern Infrastructure Using Terraform Across VMware, AWS and Public Cloud Environments - 2 days

FA-0442 · 2 days

DevOps, Cloud & Infrastructure · Cybersecurity · Automation, RPA & Power Platform

---

Introduction

Modern infrastructure teams are under constant pressure to deliver environments faster, maintain consistency across platforms, and reduce operational overhead without sacrificing governance or reliability. Terraform has become one of the most widely adopted infrastructure-as-code platforms because it allows engineers to define, provision and manage infrastructure in a repeatable and scalable way across on-premises and cloud ecosystems alike.

This intensive two-day hands-on course is designed for systems administrators, infrastructure engineers, DevOps professionals and cloud practitioners who want practical experience using Terraform in real-world environments. Rather than focusing on academic examples, the course emphasizes operational workflows commonly used in enterprise infrastructure teams. Participants will begin by configuring Terraform on Linux systems using remote SSH access and Visual Studio Code before progressing into infrastructure deployment on VMware vSphere and Amazon Web Services.

The VMware component replaces the Azure content entirely and focuses on practical vSphere automation scenarios including virtual machine provisioning, networking integration, templates, cloning operations and datastore management using Terraform. Participants will also gain exposure to modern Terraform practices including provider management, state handling, modularization and infrastructure lifecycle management aligned with current Terraform 1.x capabilities and enterprise deployment patterns.

The course is delivered by an instructor with over 30 years of industry experience, ensuring the content reflects current operational realities, enterprise expectations and production-grade implementation practices rather than purely theoretical exercises.

Learning outcomes

By the end of this course participants will be able to:

- Understand the principles of Infrastructure-as-Code (IaC) and Terraform architecture.
- Install, configure and validate Terraform within a Linux-based administrative environment.
- Use SSH and Visual Studio Code effectively for remote infrastructure automation workflows.
- Create, validate and execute Terraform configurations locally.
- Deploy and manage VMware vSphere infrastructure using Terraform.
- Provision virtual machines in VMware using templates, cloning and Terraform resource definitions.
- Configure AWS infrastructure components including VPCs, networking and EC2 instances using Terraform.
- Understand Terraform providers, modules, variables, outputs and dependency handling.
- Work with Terraform state management concepts including local and remote state backends.
- Apply Terraform operational best practices including version pinning, modularization and infrastructure lifecycle management.
- Understand contemporary Terraform ecosystem developments, governance considerations and enterprise deployment concerns.

Prerequisites

Participants attending this course should ideally have:

- Basic familiarity with Linux command-line operations.
- Ability to connect to remote systems using SSH.
- Basic understanding of virtualization technologies and VMware vSphere environments.
- General knowledge of cloud infrastructure concepts.
- Basic networking knowledge including IP addressing, subnets and firewall concepts.
- Familiarity with Visual Studio Code or similar editors.
- VMware vSphere access credentials with permissions to deploy and manage virtual machines.
- AWS credentials with sufficient permissions to provision networking and compute resources.
- Basic understanding of infrastructure concepts such as virtual machines, templates, datastores and networking.
- Familiarity with Git or version control concepts is helpful but not mandatory.

Training outline

1. Introduction to Terraform and Infrastructure Automation

Understanding Infrastructure-as-Code

1. Evolution of infrastructure management
2. Declarative versus imperative automation
3. Benefits of Infrastructure-as-Code in enterprise environments
4. Operational consistency and repeatability

Terraform Fundamentals

1. Terraform architecture
2. Terraform workflow lifecycle
3. HashiCorp Configuration Language (HCL) fundamentals
4. Providers, resources and data sources
5. Execution plans and state management concepts

Current Terraform Ecosystem and Industry Practices

1. Terraform 1.x enhancements
2. Provider ecosystem maturity
3. Governance and operational considerations
4. Infrastructure drift and compliance concerns
5. Enterprise adoption patterns

2. Linux Environment Preparation and Tooling Setup

Terraform Installation and Configuration

1. Downloading Terraform binaries

2. Linux installation procedures
3. Environment path configuration
4. Terraform version verification
5. CLI initialization and validation

Remote Administrative Access

1. SSH fundamentals
2. Key-based authentication
3. Remote Linux administration workflows
4. Secure remote access considerations

Visual Studio Code Integration

1. VS Code installation and configuration
2. Terraform extensions and plugins
3. Remote SSH integration
4. Formatting and linting support
5. Workspace organization

Terraform Command-Line Operations

1. terraform init
2. terraform fmt
3. terraform validate
4. terraform plan
5. terraform apply
6. terraform destroy
7. terraform state operations

3. Terraform Local Development and Core Concepts

Building Initial Terraform Configurations

1. Creating Terraform project structures
2. Writing basic resource definitions
3. Variables and outputs
4. Local resource provisioning exercises

Terraform Workflow Execution

1. Initialization process
2. Plan generation and interpretation
3. Applying infrastructure changes
4. Incremental modifications
5. Resource destruction workflows

Terraform State Management

1. Understanding terraform.tfstate
2. State dependencies
3. Risks of manual state manipulation
4. State locking concepts
5. Local versus remote state backends

Terraform Modularization

1. Introduction to modules
2. Root modules and child modules
3. Variable passing between modules
4. Output sharing

5. Reusable infrastructure design

Provider and Version Management

1. required_providers block
2. Version constraints
3. Provider initialization
4. Provider compatibility management
5. Dependency handling

4. Deploying Infrastructure on VMware vSphere with Terraform

VMware vSphere Provider Fundamentals

1. Overview of the vSphere provider
2. vCenter integration architecture
3. Provider authentication methods
4. Provider version management

VMware Environment Preparation

1. vCenter connectivity validation
2. Datacenter and cluster discovery
3. Resource pool identification
4. Datastore configuration
5. Virtual networking preparation

VMware Terraform Configuration Development

1. Defining provider blocks
2. Using data sources for VMware inventory

3. Working with datacenters and clusters
4. Selecting templates and datastores
5. Virtual machine resource definitions

Virtual Machine Provisioning with Terraform

1. Deploying VMs from templates
2. Cloning operations
3. CPU and memory allocation
4. Disk provisioning and datastore selection
5. Network interface configuration
6. Guest OS customization
7. VMware Tools integration

Managing VMware Infrastructure Changes

1. Scaling VM resources
2. Modifying network assignments
3. Updating VM configurations
4. Managing lifecycle operations
5. Controlled infrastructure destruction

VMware Operational Best Practices

1. Template management strategies
2. Naming standards
3. Infrastructure organization
4. State management considerations for VMware
5. Enterprise operational workflows

5. Deploying Infrastructure on AWS with Terraform

AWS Provider Fundamentals

1. AWS provider architecture
2. Provider authentication methods
3. IAM integration concepts
4. Provider versioning best practices

AWS Credential Configuration

1. IAM users and roles
2. Access key management
3. Shared credentials file usage
4. Environment variable authentication

Building AWS Infrastructure with Terraform

1. VPC creation
2. Public and private subnets
3. Route tables and gateways
4. Security groups
5. EC2 instance provisioning

Variables, Outputs and Resource Customization

1. Region selection
2. Instance sizing
3. Key pair integration
4. Resource tagging
5. Output management

Terraform Execution and Validation

1. **terraform init for AWS**
2. **Plan generation and review**
3. **Infrastructure deployment**
4. **AWS console validation**
5. **CLI-based verification**

Advanced AWS Terraform Concepts

1. **S3 bucket provisioning**
2. **Introduction to autoscaling concepts**
3. **Multi-module architectures**
4. **Networking module separation**
5. **Compute module abstraction**

AWS State and Scalability Considerations

1. **Remote state using S3**
2. **State locking with DynamoDB**
3. **Managing large Terraform environments**
4. **Performance considerations**
5. **Team collaboration strategies**

This course outline is intended as a professional training framework and guideline only. The instructor reserves the right to amend, reorganize, expand or refine the course content, sequence, lab coverage and technical focus areas as deemed necessary to accommodate audience requirements, evolving technologies, operational constraints and time considerations without prior notice.

∗Optional

© 2026 Faysal Aziz · Kuala Lumpur