Faysal Aziz

Terraform for Azure

Real-World Infrastructure on Linux

FA-0440 · 2 days

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

---

Introduction

Hands-on Terraform mastery for Azure, powered by live Linux server labs and three decades of industry wisdom - 2 days

---

In an era defined by dynamic cloud infrastructure and relentless automation, mastering Terraform on Azure isn’t just valuable—it's indispensable.

This course eschews abstract theory for tangible skills: you will configure, deploy, and manage actual infrastructure using real Linux servers, mirroring genuine enterprise workflows. Under the guidance of an instructor with over 30 years of industry experience, you’ll gain expertise rooted in real-world practice, ready to meet the demands of modern DevOps and cloud engineering roles.

Learning outcomes

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

- Set up and configure Terraform for Azure deployments from scratch
- Write and manage reusable HCL modules, variables, and outputs for Azure resources
- Implement robust Terraform state management and collaboration using Azure Blob Storage
- Design, deploy, and secure Azure infrastructure (VMs, networking, storage) using real Linux server labs
- Automate Terraform workflows with CI/CD pipelines (Azure DevOps or GitHub Actions)
- Troubleshoot Terraform configurations effectively using logs, debugging, and best practices
- Apply Terraform modules in multi-environment strategies with control over drift and remote state

- Ensure compliance, cost optimization, and security in Terraform-managed Azure environments

Prerequisites

- Comfortable working with Linux environments, including command-line tools and shell scripting
- Basic understanding of networking concepts, IP addressing, and Azure fundamentals (VMs, VNet, storage)
- Familiarity with Git or version control workflows is advantageous

Training outline

1. Infrastructure as Code & Terraform Foundations

- Principles of Infrastructure as Code within modern DevOps (GitOps, automation workflows)
- Why choose Terraform? Cloud-agnostic declarative IaC benefits
- Terraform workflow: terraform init, plan, apply, destroy
- Installing Terraform on Linux and configuring authentication with Azure (via Service Principals)

2. Terraform Configuration Language (HCL)

- Anatomy of a Terraform configuration file: providers, resources, variables, outputs
- Creating and managing Azure core resources: resource groups, VMs, storage accounts, VNets
- Modular design: writing reusable Terraform modules, code organization, inputs/outputs

3. State Management & Collaboration

- Understanding Terraform state, handling dependencies, and managing configuration drift
- Configuring remote state using Azure Blob Storage for team collaboration
- Hands-on lab: migration from local state to remote backend

4. Advanced HCL Logic & Programming Constructs

- Applying count, for_each, conditional expressions, maps, lists, functions, and data sources
- Debugging strategies with logging (e.g., TF_LOG), error handling, and state manipulation
- Lab: dynamic resource deployment, loops, and advanced constructs

5. Modular Infrastructure & Reusability

- Deep dive into modules: structure, versioning, nested modules, and publishing practices
- Refactor earlier work into well-designed modules for reuse and standardization

6. CI/CD Integration & Automation

- Automating Terraform workflows with Azure DevOps pipelines or GitHub Actions for validation and deployment
- Incorporating Terraform operations into Git-based CI/CD for multiple environments (dev/staging/prod)

7. Compliance, Cost, and Security Best Practices

- Applying best practices for Azure infrastructure: RBAC via Azure Active Directory, resource tagging, and cost monitoring
- Introduction to policy as code: Sentinel, reusable reference architectures, and security-by-design using Terraform modules

8. Real-World Linux Lab Integration

- Hands-on practice on real Linux servers: writing Terraform that spins up fully configured Azure environments
- Labs covering networking, compute, storage, module reuse, and CI/CD integration using live servers
- Simulating configuration drift resolution, state locking, and collaboration workflows in a live setting

9. Troubleshooting, Review & Next Steps

- Review of labs: diagnosing errors, fixing drift, managing state, and maintaining infrastructure
- Guidance on continued learning: Terraform Cloud, Enterprise features, and certification paths
- Course wrap-up and Q&A, with suggestions for real-world application and scaling Terraform use

---

Summary

This immersive 2-day course takes you from the foundational principles of Infrastructure as Code to the advanced mechanics of Terraform in Azure—all grounded in real Linux server labs. With more than three decades of industry insight guiding each exercise, you'll leave equipped not just with theoretical knowledge, but with actionable expertise to deploy, automate, and manage Azure infrastructure like a seasoned cloud engineer.

© 2026 Faysal Aziz · Kuala Lumpur