Faysal Aziz

Terraform Associate + AWS

Pass the exam, ship production-grade AWS, and use the latest features safely - 3 days (addon)

FA-0437 · 3 days

DevOps, Cloud & Infrastructure

Introduction

This add-on assumes you've completed **Terraform Fundamentals** (it's a firm prerequisite) and prepares you for the **HashiCorp Certified: Terraform Associate (003)** exam while building real AWS infrastructure from Linux hosts.

We explicitly map labs to the official (003) objectives (product version tested: **Terraform 1.3**) and also teach newer capabilities—testing (1.6+), test mocking (1.7), improved validations (1.9), and AWS Provider v6 changes—so your skills are current beyond the exam. You will work from real Linux servers, integrate with HCP Terraform where it makes sense, and practice production-ready patterns like module versioning, remote state, policy as code, and run tasks.

The instructor has **30+ years of industry experience** and uses **industry-demand** scenarios, not classroom-only demos.

Learning outcomes

- Confidently answer exam domains (IaC concepts, workflow, state, modules, HCP Terraform, etc.) with targeted practice tied to the official objectives.
- Build a production-style AWS stack (networking, compute, IAM) using AWS Provider **v6** and handle breaking changes/upgrade paths.
- Implement remote state on S3 with **lockfile-based** locking and bucket versioning; apply least-privilege IAM for state access.
- Use moved and import blocks to refactor and adopt brownfield AWS resources cleanly.
- Validate and test modules using terraform test, mocks, and assertions; add variable validations that reference other inputs.

- Operate with HCP Terraform: CLI-driven runs, workspaces, run tasks, and policy sets (Sentinel/OPA).

Prerequisites

- **Required:** Completion of **Terraform Fundamentals (2 days)** above.
- Linux proficiency and Git.
- An AWS account with rights to create VPC, IAM, EC2, S3 (for state).
- Optional: HCP Terraform org for collaboration.

Training outline

1. Exam-aligned quick review (003 objectives)

- IaC advantages and Terraform's purpose vs other tools; provider-agnostic model.
- Core workflow, commands, and common flags; provider installation and the lock file.
- State fundamentals: local vs remote; sensitive data; drift; backends; back-of-the-envelope threat modeling.

2. AWS Provider v6 readiness

- Reading the v6 **upgrade guide**, identifying breaking changes in resources you'll use.
- Version pinning, constraints in required_providers, and safe upgrades across teams.

3. Production-style AWS build (end-to-end lab on Linux)

Base networking module

- VPC with public/private subnets across AZs, NAT gateway strategy, route tables.
- Security groups: least privilege defaults; SG referencing patterns.

Compute & access

- EC2 module with user-data, IMDSv2, and SSM Session Manager (no SSH in prod pattern).
- IAM roles/policies for instances and CI; short-lived creds patterns.

Remote state

- S3 backend with **use_lockfile = true** and bucket **versioning**; why DynamoDB locking is **deprecated**; IAM policy for narrow state access.

Cross-stack composition

- Share outputs via terraform_remote_state; wiring ALB/ASG/target groups with module outputs.

- **Cost/safety add-ons (brief): tagging discipline, destroy-guards, and approval flows.**

4. Modules at scale

- Turning our VPC/EC2 into internal modules; semantic versioning and changelog habits.
- Registry usage (public/private), pinning with ~> vs exact, and compatibility promises.
- **Refactoring live modules with moved**; keep historical moved entries to preserve upgrade paths.

5. Brownfield adoption & import

- Write **import blocks** for existing S3 buckets, IAM roles, or EC2 instances; multi-import via for_each; documenting what was imported and why.

6. Validation & testing beyond the exam

- Variable validations and **cross-object** checks (v1.9); examples that gate dangerous inputs.
- **terraform test** deep dive: test file layout, assertions, lifecycle, and teardown; add **mocks** (v1.7) for provider calls.
- Integrate tests into CI on Linux runners; caching plugins safely.

7. Operating with HCP Terraform (Terraform Cloud)

- CLI-driven run workflow: cloud block, terraform login, workspaces, variables, and remote execution.
- **Run Tasks**: where they hook (pre/post plan/apply), advisory vs mandatory; wiring a sample run task.
- **Policy as Code** with **Sentinel/OPA**: policy sets, enforcement levels, and exempting test workspaces; quick “deny public S3” policy walkthrough.
- Agents and private module registries (overview and when to adopt).

8. Exam workflow & readiness

- Objective-to-lab mapping checklist; study plan anchored to HashiCorp’s official exam resources and sample questions.
- Question types, time management, and common distractors (e.g., overusing provisioners).
- What’s tested on 1.3 vs what you’ll see in modern Terraform (we teach both; you answer per exam).

Notes on Updates

- Latest stable at the time of writing is **Terraform v1.12**; course labs and cautions reflect its changes.
- **S3 backend** now supports **native locking** via lockfiles and marks DynamoDB locking as **deprecated** in docs—our state labs follow this modern approach.

© 2026 Faysal Aziz · Kuala Lumpur