

Terraform Foundations and AWS Deployment

Configuration, state, reusable modules and resource adoption

FA-0700 · 2 days

DevOps, Cloud & Infrastructure

Introduction

This two-day course introduces Terraform configuration and lifecycle operations through selected AWS lab resources. Participants work from basic syntax and providers to state, backends, reusable modules and existing-resource adoption.

Prepared Linux exercises provide a consistent learning environment; Terraform is not limited to Linux. Core VPC, EC2 or S3 examples are practical, while RDS and Lambda use bounded examples or demonstrations. Review plans, access, consumption and cleanup before applying infrastructure changes.

Learning outcomes

- Explain IaC and the Terraform configuration, provider and resource model.
- Run init, validate, plan, apply and destroy in a controlled exercise.
- Create and modify selected AWS resources using a reviewed configuration.
- Explain state sensitivity, remote backends and suitable locking controls.
- Use a small module, variables and resource references to model dependencies.
- Distinguish workspaces from security isolation and explain existing-resource import checks.
- Diagnose selected configuration and provider errors.

Prerequisites

- Basic understanding of cloud computing concepts.

- Familiarity with AWS and Linux command line.
- Basic knowledge of IDE usage and scripting.

An authorised AWS training account with scoped permissions, prepared tool versions and an agreed lab-resource budget/cleanup procedure.

Training outline

1. Day 1 — IaC and Terraform fundamentals

- Overview of Infrastructure as Code (IaC).
- Terraform vs. other IaC tools.
- Install and configure Terraform in the prepared Linux lab; compare supported workstation alternatives
- Containerised Terraform execution: environment, provider cache and state-access considerations

Configuration model

- Understanding Terraform syntax and structure.
- Writing your first Terraform configuration.
- Terraform providers and resources.

2. Day 1 — AWS resources and lifecycle

- Setting up AWS with Terraform.
- Creating and managing AWS resources (EC2, S3, VPC).
- State sensitivity, access controls and remote-state options for AWS

Commands and diagnostics

- Terraform init, plan, apply, and destroy.
- CLI workspaces as separate state instances, not a security-boundary substitute
- Basic debugging in Terraform.

3. Day 2 — Modules, dependencies and remote state

- RDS and Lambda configuration examples or demonstrations
- Terraform modules and reusable components.
- Resource references and explicit dependencies only where necessary

- Terraform backends for state management.

Evaluate an S3 remote backend with access controls, versioning and supported lockfile locking; do not introduce deprecated DynamoDB locking as the default.

4. Day 2 — Existing resources and review

Import a selected existing resource into state with matching configuration; review the subsequent plan before applying.

Use import blocks or suitable CLI import according to the prepared workflow. Importing state does not automatically prove complete configuration or prevent later changes.

Troubleshoot the sample, check consumption and clean up only the designated training resources.