Infrastructure as Code with Terraform

Terraform Course Content (Zero to Advanced)

Course Overview

Title: Infrastructure as Code with Terraform
Level: Beginner to Advanced
Duration: 30–40 hours
Outcome: Design, deploy, and manage production-grade infrastructure using Terraform

---

Module 1: Introduction to Infrastructure as Code (IaC)

Objective: Understand why Terraform is used

Topics

- What is Infrastructure as Code?
- Problems with manual infrastructure
- IaC tools comparison: Terraform vs CloudFormation vs ARM vs Ansible
- Declarative vs Imperative models
- Terraform use cases in real organizations

Hands-On

- Install Terraform (Windows/Linux/Mac)
- Setup AWS CLI & IAM user

Module 2: Terraform Basics

Objective: Learn Terraform fundamentals

Topics

- Terraform architecture
- Terraform workflow
- Terraform providers
- Terraform configuration files

Hands-On

- Create your first EC2 instance
- Understand execution plan

Module 3: Terraform Language (HCL)

Objective: Master Terraform syntax

Topics

- HashiCorp Configuration Language (HCL)
- Blocks, arguments, and expressions
- Resource blocks
- Data sources
- Comments & formatting

Hands-On

- Create: VPC, Subnet, Security Group
- Use data source for AMI

Module 4: Variables & Outputs

Objective: Make Terraform code reusable

Topics

- Input variables
- Variable types
- Variable precedence
- Default values

- Output values

Hands-On

- Parameterize EC2 instance size
- Output instance public IP

Module 5: Terraform State Management

Objective: Understand Terraform state deeply (important for interviews)

Topics

- What is `terraform.tfstate`
- Local vs Remote state
- State locking
- State file security
- `terraform state` commands

Hands-On

- Configure remote state using:
  - S3 backend
  - DynamoDB for locking

Module 6: Terraform Modules

Objective: Write clean, reusable infrastructure code

Topics

- What are modules?
- Root module vs child module
- Module inputs & outputs
- Module versioning
- Public Terraform Registry

Hands-On

- Create: VPC module, EC2 module
- Consume modules in root configuration

Module 7: Terraform Functions & Expressions

Objective: Write dynamic Terraform code

Topics

- Built-in functions
- Conditionals
- For expressions

Hands-On

- Dynamic security group rules
- Conditional resource creation

Module 8: Terraform with AWS (Deep Dive)

Objective: Build real AWS infrastructure

Topics

- VPC, Subnets, Route Tables
- Internet Gateway & NAT Gateway
- EC2, ALB, ASG
- RDS, S3, IAM
- ECS/EKS overview with Terraform

Hands-On

- Full 3-tier architecture:
  - VPC
  - ALB
  - Auto Scaling Group

    - RDS

Module 9: Terraform Workspaces

Objective: Manage multiple environments

Topics

- What are workspaces?
- Use cases
- Workspace limitations
- Dev / QA / Prod strategy

Hands-On

- Create dev & prod workspaces
- Deploy environment-specific resources

Module 10: Terraform Best Practices

Objective: Write production-ready Terraform

Topics

- Folder structure
- Naming conventions
- DRY principles
- Version pinning
- Secrets management
- Terraform formatting & linting

Module 11: Terraform Security

Objective: Secure Terraform deployments

Topics

- Sensitive variables
- Secrets management:
    - AWS Secrets Manager
    - SSM Parameter Store
- IAM least privilege
- Prevent accidental deletion

Hands-On

- Store DB password securely
- Mark sensitive outputs

Module 12: CI/CD with Terraform

Objective: Automate Terraform deployments

Topics

- Terraform in CI/CD
- Terraform with:
    - GitHub Actions
    - Azure DevOps

Hands-On

- CI pipeline:
    - Validate
    - Plan
    - Apply (manual approval)

Module 13: Terraform Troubleshooting

Objective: Debug Terraform issues

Topics

- Common Terraform errors
- Provider issues

- State corruption
- Drift detection
- Debug logs (`TF_LOG`)

Hands-On

- Fix broken state
- Import existing resources

Module 15: Capstone Project

Objective: Apply everything learned

Project

Build Production-Ready AWS Infrastructure

- Modular Terraform code
- Remote backend
- Multi-environment support
- CI/CD pipeline
- Security best practices