Enterprise Infrastructure Engineering with Terraform (Azure) Training

Duration: **4 Day**

BILGINÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Enterprise Infrastructure Engineering with Terraform (Azure) Training is a **four-day advanced course** designed for experienced professionals who already work with **Terraform in Azure production environments** and need to manage infrastructure safely within established **DevOps delivery pipelines**.

As the **third course in the Terraform learning journey**, the programme moves beyond writing Terraform configurations and preparing them for reuse and automation.

While earlier training focuses on **Terraform development**, **modular configuration** and **automation**, this course examines the operational challenges that emerge once Terraform becomes part of **CI/CD workflows**.

These challenges are particularly important in organisations where infrastructure is shared between teams, existing resources must be managed, and changes require **review, approval and clear justification**.

Participants learn to approach Terraform not simply as an **infrastructure deployment tool**, but as a **state-driven operating model** integrated into DevOps practices.

This perspective emphasises how organisations manage **resource ownership**, **change intent** and **operational risk** throughout the infrastructure lifecycle.

The course reflects the way Terraform operates in mature enterprise environments, where **automation carries out approved actions while people remain responsible for the decisions behind them**.

Rather than focusing only on **Terraform and CI/CD integration**, the programme explores how Terraform should be governed and operated as part of a wider DevOps delivery model.

The training is built around several key principles:

- **Pipelines execute changes, but people make the decisions.**
- **Terraform plans** are treated as **governance artefacts** that require review.
- **Terraform State** represents boundaries of **ownership and authority**.
- **Infrastructure changes** should be intentional and controlled rather than automatically accepted.

These principles make the course particularly relevant to organisations managing **regulated, multi-team Azure environments**.

Throughout the programme, learners reinforce their knowledge through **hands-on laboratories** based on realistic **Azure and DevOps operating models**.

Prerequisites

Enterprise Infrastructure Engineering with Terraform (Azure) Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

This is an **advanced-level course**.

Participants should already be confident using **Terraform in Azure** and have a working understanding of **DevOps delivery concepts**.

To gain the greatest benefit from the programme, learners are encouraged to have completed:

- **Terraform in Practice**
- **Intermediate Terraform Cloud Automation**

Alternatively, equivalent practical experience is acceptable, particularly in the following areas:

- Working with **Terraform modules**
- Managing **remote state**
- Running Terraform through **CI/CD pipelines**

Who Should Attend

This course is intended for experienced technical professionals involved in managing enterprise Azure infrastructure through Terraform and DevOps practices.

It is particularly suitable for:

- **Systems Engineers**
- **Site Reliability Engineers (SREs)**
- **DevOps Engineers**

Enterprise Infrastructure Engineering with Terraform (Azure) Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By completing the **Enterprise Infrastructure Engineering with Terraform (Azure) Training**, participants will develop the skills needed to manage Terraform safely within complex, governed Azure environments.

By the end of the course, learners will be able to:

- **Operate Terraform securely across multiple Azure environments** using DevOps pipelines.
- Understand **Terraform State** as a boundary defining **ownership and authority**.
- Analyse **Terraform plans generated by CI/CD** to evaluate potential risks and impacts.
- **Refactor live Terraform configurations** without causing unintended infrastructure destruction.
- Design and consume **reusable Terraform modules** that account for governance requirements.
- Implement **intent-driven environment promotion and approval workflows**.
- Integrate Terraform into **governed DevOps delivery models**.
- Correctly consume **centrally managed Azure Landing Zones**.
- Reconcile **existing workloads** with modern enterprise platform standards.

This course enables participants to manage**Terraform as an enterprise operating model**, bringing together **automation, accountability, change governance and risk management** within established Azure DevOps delivery processes.

Outline

The programme is organised into concept-led modules, each supported by**practical laboratory exercises** reflecting real-world **Azure infrastructure and DevOps operations**.

Module 0: Course Philosophy – Terraform in DevOps-Driven Organisations

This introductory module establishes the approach required to operate**Terraform as part of a DevOps delivery system**, rather than treating it as an independent deployment utility.

Participants consider the relationship between **automation, human decision-making, accountability and governance** in Terraform operations.

Module 1: Terraform State and Resource Identity

This module examines how Terraform determines**resource identity through State** and how this

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

influences infrastructure lifecycle behaviour in automated pipelines.

Participants explore the connection between **resource identity**, **state management** and the effects of changes executed through **CI/CD workflows**.

Module 2: Input Hygiene, Validation, and Determinism

This module introduces **defensive Terraform configuration practices** designed to prevent unsafe inputs and configurations from entering Terraform State through CI/CD.

The focus is on **input validation**, **configuration safety** and achieving **deterministic, predictable outcomes**.

Module 3: Dynamic Configuration and Predictable Scaling

Participants explore how to expand and scale **Terraform configurations** while avoiding unexpected resource changes during automated deployments.

The module focuses on using **dynamic configuration techniques** without introducing hidden infrastructure churn.

Module 4: Multi-Environment State and Remote Backends

This module covers the operational considerations involved when Terraform manages multiple Azure environments through shared delivery pipelines.

Key areas include **remote state management**, **environment isolation**, **remote backends** and the risks associated with shared CI/CD execution.

Module 5: Modular Design and Safe Refactoring

This module focuses on evolving existing Terraform configurations while preserving **resource identity** in live DevOps environments.

Participants examine approaches to **safe refactoring** that reduce the risk of unintended resource destruction or disruption.

Module 6: Reuse, Contracts, and Organisational Boundaries

This module explores the use of **shared Terraform modules** across different teams and organisational boundaries.

Participants examine **module consumption**, **interface contracts** and the management of changes that affect multiple teams.

Module 7: Intent-Driven Promotion and Governance

This module examines how **environment promotion**, **approvals** and **governance controls** should operate within DevOps workflows.

Participants learn why promotion and approval should be treated as **explicit controls based on**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

change intent, rather than implicit steps within automation.

Module 8: Consuming and Reconciling with a Centralised Landing Zone

The final module addresses the challenges of working with**existing infrastructure (brownfield environments)** alongside centrally managed**Azure Landing Zones**.

Participants explore how to consume shared platform capabilities correctly and reconcile existing workloads with **governed DevOps practices** and **modern platform standards**.