

Terraform With Azure

The **Terraform with Azure** is designed to Master Azure infrastructure automation using Terraform with hands-on, real-world concepts from basics to advanced.

Who It's For

- Beginners who want to start **Infrastructure as Code** using Terraform on AWS
- Cloud learners aiming to automate **Azure resource** creation
- System admins and **DevOps engineers** looking to improve automation skills
- Anyone preparing for **real-time cloud project** implementation

What You'll Learn

- Core concepts of **Terraform** and how it works with **Azure**
- Creating and managing Azure resources like storage accounts Terraform
- Handling state files, remote backends, and best practices
- Writing reusable and structured Terraform code using variables and modules

- Advanced concepts like lifecycle rules, meta-arguments, dynamic blocks, and functions

Benefits You'll Gain

- Ability to automate Azure infrastructure deployment confidently
- Hands-on skills aligned with **real-world DevOps** practices
- Faster and error-free infrastructure management
- Strong foundation for DevOps, Cloud, and Terraform-related roles

Course Details

- **Duration:** 20 days (1 hour per day)
- **Mode:** Fully online – attend live
- **Start Date:** Flexible enrollment – begin anytime!

Topics that will be covered

Introduction to Terraform

- ◆ Understanding Infrastructure as Code
- ◆ Why we need Infrastructure as Code
- ◆ What is Terraform and its benefits
- ◆ Challenges with the traditional approach
- ◆ Terraform workflow
- ◆ Installing Terraform

Terraform Providers

- ◆ Terraform Providers

- ◆ Provider version vs Terraform Core Version
- ◆ Why Terraform Version matters
- ◆ Version Constraints
- ◆ Operators for Versions

Azure Connect

- ◆ Authentication and Authorization Azure
- ◆ Creating Principal account in Azure
- ◆ Use Access keys to authorize Terraform
- ◆ Understanding Visual Studio Code
- ◆ Terraform basic Commands
 - Init
 - Validate
 - Plan
 - Apply
 - Destroy

Azure Storage account Management

- ◆ Storage account Management using Terraform
- ◆ Deploy Storage account
- ◆ Delete Storage account
- ◆ Understand the process behind the Terraform

State file Management and Remote Management

- ◆ How Terraform updates infrastructure
- ◆ Terraform state file
- ◆ State file best practices
- ◆ Remote backend setup with Storage account

- ◆ Storage account locking file
- ◆ State Management

Terraform Variables

- ◆ Variables of Terraform
 - Input Variables
 - Local Variables
 - Output Variables

File Structure

- ◆ Terraform file organization
- ◆ Sequence of file loading
- ◆ Best practices for file loading
- ◆ Code organization patterns

Types of Constraints

- ◆ String, Numbers, Bool types
- ◆ Map, Set, List, Tuple, Object types
- ◆ Type validation and constraints
- ◆ Complex type definitions

Terraform Meta Arguments

- ◆ Understanding all Terraform meta-arguments
 - Count
 - For_each
 - Depends_on
 - Lifecycle
 - Provider
- ◆ Output transformations with **for** expressions
- ◆ Best practices for each meta-arguments

Terraform Lifecycle Meta-arguments (AWS)

- ◆ Create_before_destroy
- ◆ Prevent_destroy
- ◆ ignore_changes
- ◆ Replace_triggered_by
- ◆ Pre condition
- ◆ Post condition

Terraform Features

- ◆ Dynamic Blocks
- ◆ Conditional Expressions
- ◆ Dynamic Blocks
- ◆ Splat Expressions

To Join this Course Contact or Ping on Whatsapp : **93427 37660**