

# Terraform With AWS

The **Terraform with AWS** is designed to Master AWS 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 **AWS 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 **AWS**
- Creating and managing AWS resources like S3 using 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 AWS 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

## **AWS Connect**

- ◆ Authentication and Authorization AWS
- ◆ Creating User and Access keys
- ◆ Use Access keys to authorize Terraform
- ◆ Understanding Visual Studio Code
- ◆ Terraform basic Commands
  - Init
  - Vilate
  - Plan
  - Apply
  - Destroy

## **AWS S3 Management**

- ◆ S3 Management using Terraform
- ◆ Deploy S3 Bucket
- ◆ Delete S3 Bucket
- ◆ 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 S3
- ◆ S3 Native state locking

- ◆ 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**

EMPOWERING THE SUCCESS WITH AI