Technical Guftgu

(Established under Ministry of MSME, Govt. of India)

Contact- +91-9870663188 or 8527556109

Website: **www.technicalguftgu.in**

- ✓ **Certificate provided**
- ✓ **Recordings Provided**
- ✓ **Classes In Hindi/urdu**
- ✓ **Expert Trainers**

Terraform With Azure (in Depth)

Prerequisites

- A foundational understanding of IT infrastructure.
- Knowledge of Unix/Linux operating system.
- Basic knowledge about Software Development Life Cycle.

Learning outcomes

Learn Terraform in depth

70+ labs helping engineers to be productive on Terraform from Day 1

Terraform Setting and Provider Block

Step-01: Terraform Settings Block Introduction
Step-02: Understand required_version in Terraform Block
Step-03: Terraform Provider Introduction
Step-04: Understand required_providers in Terraform Block and Provider Block
Step-05: Terraform Apply and Destroy Commands Auto Approve Option

Terraform Multiple Provider

Step-01: Terraform Multiple Providers Introduction
Step-02: Implement Terraform Multiple Providers & Clean-Up

Terraform Dependency Lock File

Step-02: Review c1-versions.tf, Create RG and Random String Resource
Step-03: Create Terraform Storage Account Resource
Step-04: Dependency Lock File Demo and CleanUp

Terraform Resource Syntax and Behavior

Step-01: Terraform Resource Syntax Introduction
Step-02: Create TF Config for Virtual Network
Step-03: Create TF Config for Subnet, Public IP and Network Interface
Step-04: Terraform Resource Behavior Introduction
Step-05: Resource Behavior: Create Resource Demo
Step-06: Understand Terraform State in detail
Step-07: Resource Behavior: Update-In-Place, Destroy-Recreate and Destroy Demo's
Step-08: Understand Terraform Desired and Current States and CleanUp

Terraform meta argument depend_on

Step-00: All Meta-Arguments Introduction
Step-01: Introduction to Meta-Argument depends_on
Step-02: Execute TF Commands without depends_on and understand Terraform Behavior
Step-03: Execute TF Commands with depends_on and understand Terraform Behavior

Provision Azure Linux vm with file and filebase64

Step-01: Introduction to Azure Linux VM using Terraform
Step-02: Create TF Config for Azure Linux VM
Step-03: Review CloudInit file for custom_data and filebase64 function
Step-04: Execute TF Commands, Verify and Clean-Up Azure All resources

Terraform meta argument with Element Function and Splat Expression

Step-01: Introduction to Meta-Argument count
Step-02: Meta-Argument Count - Azure Resource Group Demo
Step-03: Introduction to Meta-Argument count for Azure Linux VM
Step-04: Learn Terraform Element, Length Functions and Splat Expressions
Step-05: Apply Element Function, Splat Expression to VMNIC and Linux VM
Step-06: Execute Terraform Commands, Verify 2 Linux VMs and CleanUp

Terraform meta argument for_each with Maps and set of string

Step-02: Meta-Argument for_each with Maps Demo
Step-03: for_each - Set of Strings Introduction and Terraform Console Command
Step-04: Implement for_each with set of strings
Step-05: for_each chaining Introduction and Review TF Configs
Step-06: Implement for_each chaining

Meta argument lifecycle create_before_destroy, Prevent Destroy and Ignore_change

Step-02: Explore default resource behavior - Delete and Recreate Resource
Step-03: Lifecycle Meta-Argument create_before_destroy demo
Step-04: Lifecycle Meta-Argument prevent_destroy demo
Step-05: Without Lifecycle Meta-Argument ignore_changes understand
Step-06: Lifecycle Meta-Argument ignore_changes demo

Terraform Input Variable

Step-01: Define Terraform Input Variable
Step-02: Terraform Input Variable Basics Demo
Step-03: Terraform Input Variables - Assign When Prompted demo
Step-04: Terraform Input Variables - CLI Argument -var
Step-05: Terraform Input Variables - CLI Argument -var by generating a TF Plan f
Step-06: Terraform Input Variables - Override with Environment Variables
Step-07: Terraform Input Variables - Override with terraform.tfvars
Step-08: Terraform Input Variables - anyfilename.tfvars with -var-file argument
Step-09: Terraform Input Variables - understand .auto.tfvars
Step-10: Terraform Input Variables - Review TF Configs for List Item
Step-11: Terraform Input Variables - Create List Variable, Verify and CleanUp
Step-12: Terraform Input Variables - Review TF Configs for Maps Item
Step-13: Terraform Input Variables - lookup function
Step-14: Terraform Input Variables - Create Map Variables, Verify and CleanUp
Step-15: Terraform Functions: Length, Substring, Lower, Upper and Contains
Step-16: Terraform Input Variables - Validation Rules with OR and contains funct
Step-17: Terraform Input Variables - Validation Rules with regex and can functio
Step-18: Terraform Input Variables - Sensitive Introduction
Step-19: Terraform Input Variables - Define Sensitive, bool and Number Variables
Step-20: Terraform Input Variables - Create Azure MySQL Server Resources
Step-21: Terraform Input Variables - Create Azure MySQL DB, Test and CleanUp
Step-22: Terraform Input Variables - Structural Type Object Introduction

Step-23: Terraform Input Variables - Create TF Configs of ST Object
Step-24: Terraform Input Variables - Execute TF Commands, Verify and CleanUp ST
Step-25: Terraform Input Variables - Create TF Configs for ST Tuple
Step-26: Terraform Input Variables - Run TF Plan and Verify tuple var value repl
Step-27: Terraform Input Variables - Introduction to Collection Type set
Step-28: Terraform Input Variables - Review TFConfigs for CT Set
Step-29: Terraform Input Variables - Execute TF Commands, Verify and CleanUp CT

Terraform Output Values

Step-00: Output Values Introduction
Step-01: Create Basic Output Values and Review TF Configs
Step-02: Execute TF Commands, Verify and learn about "terraform output" command
Step-03: Output Values with Sensitive flag and also "terraform output -json"
Step-04: Output Values with Meta-Argument count and Splat Expression
Step-05: Output Values with Meta-Argument for_each and For Expression - Introduc
Step-06: Create List Outputs
Step-07: Create Map Outputs and use key and values functions

Terraform Local values

Create Local Values Terraform Config

Terraform Conditional Expression

Step-01: Terraform Conditional Expressions Introduction and Create TF Configs
AZHCTA-36-02-TFCE-Conditional-Expressions-Execute-TFCommands-Verify-CleanUp
AZHCTA-36-03-TFCE-Conditional-Expressions-in-a-Resource-Demo

Terraform Data Source

Step-02: Create Datasource for Resource Group Resource
Step-03: Create Datasource for Virtual Network
Step-04: Create Datasource for Azure Subscription

Terraform Remote State and Locking

Step-02: Create Azure Storage Account and Container
Step-03: Create TF Backend Block with Azure Storage Account and Review TF Config
Step-04: Execute TF Commands, Verify Remote State Storage and Locking Features
Step-05: Understand Azure Storage Account TF State File Versioning and CleanUp

Terraform state command

Step-01: Terraform Show Command to read Terraform Plan Files
Step-02: Terraform Show Command to read Terraform State Files
Step-03: Terraform State List and Show Commands
Step-04: Terraform State mv command
Step-05: Terraform State rm command and replace-provider command
Step-06: Terraform State Push Pull and Force-Unlock Commands
Step-07: Terraform Taint and Untaint Commands
Step-08: Terraform Plan and Apply - "-target" option for Resource Targeting
Step-01: Introduction to "terraform apply -refresh-only" command
Step-02: Execute TF Commands with "terraform apply -refresh-only" and Clean-Up

Terraform CLI with Workspace

Step-02: Review TF Configs and understand terraform.workspace variable
Step-03: Create Resources in default workspace and learn commands workspace list
Step-04: Create new workspace, create resources and understand state files

Step-05: Learn to delete resources in workspaces and deleting workspaces
Step-06: Implement CLI Workspaces with Remote State Storage Backend

Terraform Provisioner

Step-01: Understand File Provisioner, Self Object and Create Connection Block
Step-02: Understand Creation-Time Provisioner and Create File Provisioners
Step-03: Execute TF Commands and Verify Files provisioned to Linux VM
Step-04: Provisioners on_failure = continue or fail verify and cleanup
Step-05: Remote-exec Provisioner Demo
Step-06: Local-exec Provisioner Demo
Step-01: Understand Null and Time Resources and Create Time Resource
Step-02: Create Null Resource, File and remote-exec Provisioners and Triggers
Step-03: Execute TF Commands, Verify Static Content and Understand more about null
Step-01: Understand Terraform Import and Import Resource Group
Step-02: Create RG Resource by referring TFSTATE file, Verify and CleanUp

Terraform Module

Step-01: Understand Terraform Modules and its features
Step-02: Create VNET Module and reference it in VMNIC Resource
Step-03: Execute TF Commands, Verify VNET, Subnet and Access Sample App
Step-04: Taint Child Module Resources and Clean-Up

Step-01: Understand about Child Modules and Create it
Step-02: Create Root Module TF Configs
Step-03: Execute TF Commands, Verify and CleanUp Static Website created
Step-04: Understand Terraform get command

Step-01: Create Git Repo and Commit Static Website TF Module Files and Create 1.
Step-02: Publish the Module to Terraform Public Registry and Verify
Step-03: Create Root Module and call public registry module newly published
Step-04: Learn Module Management in Public Registry and Module Versioning

Step-01: Introduction to Terraform Private Module Registry
Step-02: Create Git Repo and Publish 1.0.0 release
Step-03: Create Github Oauth Connection and Publish Private Module in TF Cloud
Step-04: Create TF Configs, Use Source as Private Module and Execute TF Commands
Step-05: Build Azure Image Builds with Packer
Step-06: Terraform with docker
Step-07: Terraform with Kubernetes
Step-08: Terraform with testing tools like selenium
Step-09: Create a complete infrastructure with Terraform and implement CI/CD in it.

KEY HIGHLIGHTS OF THIS TRAINING PROGRAM:

- ✓ *Entire training programme is in Hindi Language for Better understanding.*
- ✓ *Special focus on Non technical and Fresher candidates.*
- ✓ *Resume Preparation for Fresher's and Experienced Both.*