Course Outline

ProTech Professional Technical Services, Inc.

Terraform 101 - Infrastructure as Code (IaC)

Course Summary

Description

As enterprises seek to deploy and maintain increasingly complex cloud infrastructure, there is a necessity to use “Infrastructure as Code” (IaC) tools, like Terraform. An open- source, state management tool developed by HashiCorp, Terraform allows developers to use a common coding interface to work through their various clouds safely and efficiently. Attendees will leave being able to write and understand Terraform code (HCL), have a clear understanding of Terraform’s various components and supporting tools, as well as when to reach for Terraform over another IaC tool, such as Ansible.

Objectives

At the end of this course, students will be able to:

- Understanding IAC
- Writing Terraform HCL code
- Core Terraform Workflow
- Deploying into common clouds such as AWS, Azure, Docker, Kubernetes, and VMWare
- Implementing and Maintaining State
- Where Terraform fits in the Enterprise CI/CD model
- Differences between Terraform and Ansible
- Best practices

Topics

- Understanding IAC
- Up and Running with Terraform
- Syntax
- Resources
- Variables and Output
- Interacting with Terraform Modules
- Terraform Templates
- Expressions
- Functions
- Implementing and Maintaining State
- CICD Piplines with Terraform
- Enterprise Case Studies
- Beyond Basics

Audience

This course is designed for:

- DevOps Engineers
- Software Developers
- Technical Managers and Leads
- System and Cloud Administrators
- Network Engineers and Developers

Prerequisites

Although not required, students with some experience programming, or pre-existing knowledge of cloud architecture, will most appreciate the technical nature of this hands-on course.

Duration

Three days

Course Outline

Terraform 101 - Infrastructure as Code (IaC)

Course Outline

I. Understanding IAC

II. Up and Running with Terraform
- A. Terraform Overview
- B. Defining “declarative”
- C. How to think about Terraform (versus Ansible)
- D. Reviewing the Terraform Configuration
- E. Running the Terraform Configuration
- F. Provisioners

III. Syntax
- A. “Low Level” HCL syntax
- B. Style Conventions
- C. Comments
- D. Blocks
- E. Arguments
- F. JSON Configuration Syntax

IV. Resources
- A. Meta-Arguments
- B. depends_on
- C. count
- D. for_each
- E. provider
- F. lifecycle
- G. Data Sources

V. Variables and Output
- A. Input Variables
- B. Output Values
- C. Local Values

VI. Interacting with Terraform Modules
- A. Module Blocks
- B. Module Sources
- C. Meta Arguments

VII. Terraform Templates
- A. templatefile Function
- B. Template Demonstration
- C. Introducing Data Sources
- D. Creating an External Data Source
- E. Building tftpl template files

VIII. Expressions
- A. Types and Values
- B. Strings and Templates
- C. Reference to Values
- D. Operators
- E. Function Calls
- F. Conditionals
- G. For Expressions
- H. Splat Expressions
- I. Dynamic Blocks
- J. Type Constraints
- K. Version Constraints

IX. Functions
- A. String
- B. Collection
- C. Numeric
- D. Encoding
- E. Filesystem
- F. Date and Time
- G. Hash and Crypto
- H. IP Network
- I. Type Conversion

X. Implementing and Maintaining State
- A. Understanding the importance of state
- B. State storage and locking
- C. importing existing resources
- D. Remote State
- E. What to do when local state is lost

XI. CICD Piplines with Terraform
- A. Terraform and GitLab pipelines
- B. Terraform and Jenkins pipelines

XII. Enterprise Case Studies
- A. Terraform and Docker
- B. Terraform and Kubernetes
- C. Terraform and Amazon AWS
- D. Terraform and Azure
- E. Terraform and VMWare
- F. Understanding how to apply Terraform to your unique infrastructure

XIII. Beyond Basics
- A. Intro to Go Programming
- B. Understating Terraform Cloud Capabilities
- C. Additional HashiCorp Offerings
- D. Backends
- E. Secrets

Course Outline

Terraform 101 - Infrastructure as Code (IaC)

Course Outline (cont'd)

Hands On Labs

- Overview of Terraform
  - LECTURE - Introduction to Terraform
  - Terraform Install
- Docker
  - Up and Running with Terraform
  - Terraform Variables
  - Output Values
  - CHALLENGE - Terraform and Docker
- Beyond Basics
  - Terraform Language Values
  - Terraform Local Action Provider
  - Terraform Expressions and Errors
  - Dynamic Operations with Functions
  - Dynamic Provisioning with tfvars Files
  - Importing Terraform Configuration
- AWS
  - Terraform and AWS
  - Output Values and AWS
  - Correcting Resource Drift and AWS
  - CHALLENGE - Terraform and AWS
- VMWare
  - Terraform and VMWare
- Terraform and Enterprise
  - Terraform Cloud
  - Deploy a Go RESTful API microservice with Terraform
  - Terraform and Ansible