

**Terraform Cloud**

**Course Number:** TRFM-102
**Duration:** 2 days

**Overview**

Accelebrate's Terraform Cloud training course introduces attendees to the advantages of developing “Infrastructure as Code” with Terraform using the Terraform Cloud SaaS product. Attendees learn how to manage the deployment of different infrastructure types, including virtual machines, containers, storage, networking, and policies via the appropriate Provider plugin. In addition, students discover how the Terraform Cloud console provides a common administration portal for managing complex multi-cloud environments across multiple workspaces.

**Note:** This online 2-day course can also be taught as 4 half-day sessions. In addition, the class can be adapted to other AWS services or cloud providers.

**Prerequisites**

Students must be at ease working at the command-line, understand Amazon Web Services (AWS) such as EC2 or S3, and have experience using an SSH Client, such as OpenSSH on Linux, macOS or WSL, or PuTTY on Windows.

**Materials**

All Terraform training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* Visual Studio Code (optional)
* Terraform Plugin for VS Code (optional)
* AWS CLI for your platform (Windows, Mac, or Linux) (optional)
* AWS access and secret key can be provided by you or by Accelebrate

**Objectives**

* Use Terraform Cloud web interface for standing up various AWS resources
* Apply plans as part of the Terraform workflow
* Use basic and complex variable types and control structures
* Improve configurations by writing reusable code modules
* Find information about other AWS modules and data sources
* Use the Terraform command-line, if desired, to interact with Terraform Cloud
* Adopt best practices

**Outline**

* Introduction to Infrastructure as Code
	+ Infrastructure as Code, Config Management, Idempotence
	+ Terraform & Terraform Cloud
	+ Introduction to Terraform Cloud
	+ Local installation of the Terraform command-line tool
* Terraform Cloud Workflow
	+ The plan
	+ Applying and re-applying plans
	+ Destroying resources
	+ Various sub-commands
* HCL Configurations
	+ Providers
	+ Variables
	+ Resources
* Variable Types
	+ Variables, passing values to the configuration, Locals
	+ Basic and complex types
* Templates
	+ Control Structures
	+ Templates
* Data Sources
	+ Provider Data Sources
* State
	+ Terraform client use of Local State
	+ Terraform Cloud “remote state” and workspaces for collaborating in teams
	+ Rolling back state to a previous revision
* Modules
	+ Modules structure
	+ Using modules
	+ Terraform Registry
	+ Writing your own modules
* Provisioners
	+ Local-exec
	+ File
	+ Remote-exec
* Importation of Foreign Resources
	+ Importation of resources created outside Terraform
		- terraform import
		- terraforming
* Auto-Scaling & Load-Balancing
	+ AWS EC2 ASG - Autoscaling Groups
	+ AWS EC2 ALB - Application Load Balancer
* AWS EKS Cluster (Elastic Kubernetes Service)
	+ Standing up an AWS EKS (Kubernetes) Cluster
	+ Managing day2 operations with Terraform
	+ Using Terraform to manage EKS resources
* Other AWS Resources
	+ Lambda
	+ VPC, EIP
	+ S3, EBS
	+ IAM, RDS
* Best Practices
	+ Terraform Best practices
	+ 3rd-party Tooling: Linters, scanners, testers
* HashiCorp Terraform Eco-System
	+ Terraform Enterprise
	+ CDK
	+ Waypoint, Boundary
	+ Certification
* Conclusion