Terraform Authoring and Operations Professional

Duration: 4 days (8hrs/day)

Prerequisites: This exam assesses both advanced configuration authoring and a deep understanding of Terraform workflows. Earning this credential reflects that the practitioner has professional-level expertise in both domains. We strongly recommend that you pass the associate-level Terraform exam before taking the professional-level exam. The professional-level exam is intended for practitioners who have extensive experience with Terraform in a production environment, and understand the concepts covered in the associate exam.

- Experience using the Terraform AWS Provider in a production environment
- HashiCorp Certified: Terraform Associate Certification (recommended)
- Linux skills, such as the ability to list and edit files via command terminal
- Experience using cloud credentials
- Familiarity with YAML, JSON, HCL, and CSV formats
- Understanding of the networking stack and networking protocols, including TCP/IP and UDP
- Advanced configuration authoring and a deep understanding of Terraform workflows

Course Objective: The Terraform Authoring and Operations Professional exam is a lab-based exam for Cloud Engineers focused on developing Terraform configuration and using it to manage infrastructure over time. You are well-qualified to take this exam if you hold the Terraform Associate Certification (or equivalent knowledge), have extensive production experience with Terraform Community Edition, Terraform Enterprise, or HCP Terraform, develop modules and organization standards to enable other users, and use Terraform to manage your own infrastructure resources, as well as enable others with varying levels of Terraform expertise to use Terraform effectively.

Cloud Platform: AWS

Lab Requirement: Participant AWS Account Required

Course Outline

1	**Manage resource lifecycle**
1a	Initialize a configuration using `terraform init` and its options

1b	Generate an execution plan using `terraform plan` and its options
1c	Apply configuration changes using `terraform apply` and its options
1d	Destroy resources using `terraform destroy` and its options
1e	Manage resource state, including importing resources and reconciling resource drift
2	**Develop and troubleshoot dynamic configuration**
2a	Use language features to validate configuration
2b	Query providers using data sources
2c	Compute and interpolate data using HCL functions
2d	Use meta-arguments in configuration
2e	Configure input variables and outputs, including complex types
2f	Analyze best practices for managing sensitive data, such as using Vault for secrets Management
3	**Develop collaborative Terraform workflows**
3a	Manage the Terraform binary, providers, and modules using version constraints
3b	Configure remote state
3c	Use the Terraform workflow in automation
3d	Share data across configurations and workspaces
4	**Create, maintain, and use Terraform modules**
4a	Create a module
4b	Use a module in configuration
4c	Refactor a module and use module versioning
4d	Refactor an existing configuration into modules
5	**Configure and use Terraform providers**
5a	Understand Terraform's plugin-based architecture

5b	Configure providers, including aliasing, versioning, sourcing, and managing upgrades
5c	Manage provider authentication
5d	Troubleshoot provider errors
6	**Collaborate on infrastructure as code using HCP Terraform**
6a	Analyze the HCP Terraform run workflow
6b	Understand HCP Terraform workspaces and their configuration options, including access
6c	Manage provider credentials in HCP Terraform
6d	Analyze policy as code and governance features