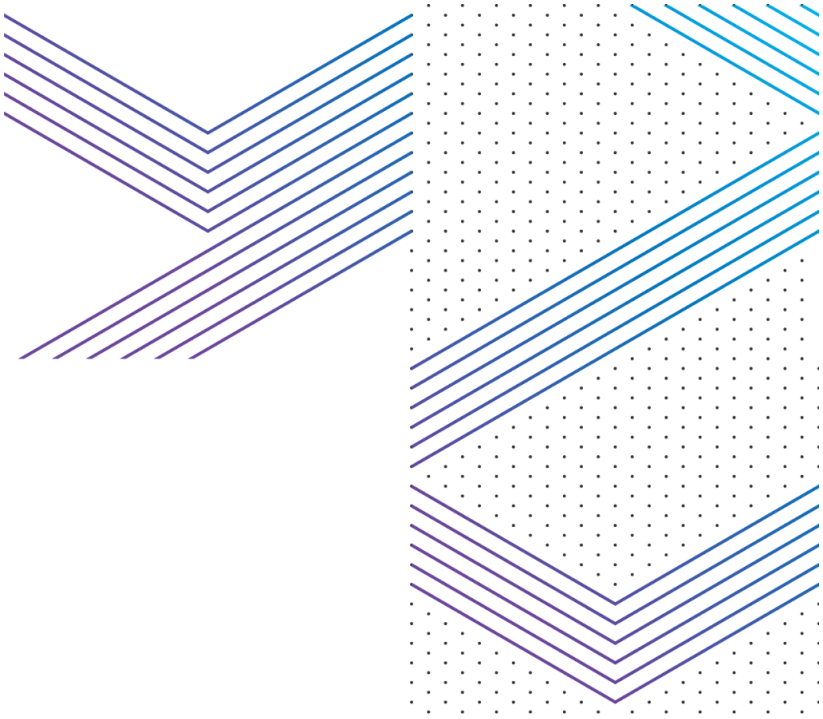

HashiCorp Certified: Terraform Associate (003)

Credential validity and objective overview

HashiCorp
an IBM Company

HashiCorp Certified: Terraform Associate (003)

Credentials (badge and certificate) are valid until their stated expiration date

The HashiCorp Certified: Terraform Associate (003) exam version was retired on January 7, 2025. However, the credentials associated with this exam are still a valid indication of one's certification status until their expiration date.

Terraform Associate 003 exam objectives

#	Objective description
1	**Understand infrastructure as code (IaC) concepts**
1a	Explain what IaC is
1b	Describe advantages of IaC patterns
2	**Understand Terraform's purpose (vs other IaC)**
2a	Explain multi-cloud and provider-agnostic benefits
2b	Explain the benefits of state
3	**Understand Terraform basics**
3a	Install and version Terraform providers
3b	Describe plugin-based architecture
3c	Write Terraform configuration using multiple providers
3d	Describe how Terraform finds and fetches providers
4	**Use Terraform outside the core workflow**
4a	Describe when to use `terraform import` to import existing infrastructure into your Terraform state
4b	Use `terraform state` to view Terraform state

4c	Describe when to enable verbose logging and what the outcome/value is
5	**Interact with Terraform modules**
5a	Contrast and use different module source options including the public Terraform Module Registry
5b	Interact with module inputs and outputs
5c	Describe variable scope within modules/child modules
5d	Set module version
6	**Use the core Terraform workflow**
6a	Describe Terraform workflow (Write → Plan → Create)
6b	Initialize a Terraform working directory (`terraform init`)
6c	Validate a Terraform configuration (`terraform validate`)
6d	Generate and review an execution plan for Terraform (`terraform plan`)
6e	Execute changes to infrastructure with Terraform (`terraform apply`)
6f	Destroy Terraform managed infrastructure (`terraform destroy`)
6g	Apply formatting and style adjustments to a configuration (`terraform fmt`)
7	**Implement and maintain state**
7a	Describe default `local` backend
7b	Describe state locking
7c	Handle backend and cloud integration authentication methods
7d	Differentiate remote state back end options
7e	Manage resource drift and Terraform state
7f	Describe `backend` block and cloud integration in configuration
7g	Understand secret management in state files
8	**Read, generate, and modify configuration**
8a	Demonstrate use of variables and outputs
8b	Describe secure secret injection best practice
8c	Understand the use of collection and structural types
8d	Create and differentiate `resource` and `data` configuration

8e	Use resource addressing and resource parameters to connect resources together
8f	Use HCL and Terraform functions to write configuration
8g	Describe built-in dependency management (order of execution based)
9	**Understand HCP Terraform capabilities**
9a	Explain how HCP Terraform helps to manage infrastructure
9b	Describe how HCP Terraform enables collaboration and governance