HashiCorp Certified: Terraform Associate (002)

<https://www.hashicorp.com/certification/terraform-associate>

The Terraform Associate certification is for Cloud Engineers specializing in operations, IT, or development who know the basic concepts and skills associated with open source HashiCorp Terraform. Candidates will be best prepared for this exam if they have professional experience using Terraform in production, but performing the exam objectives in a personal demo environment may also be sufficient. This person understands which enterprise features exist and what can and cannot be done using the open source offering. Visit our exam partner to [schedule and take the exam](https://hashicorp-certifications.zendesk.com/hc/en-us/articles/360049382552).

**Prerequisites**

* Basic terminal skills
* Basic understanding of on premises and cloud architecture

**Product Version Tested**

Terraform 1.0 and higher.

**Preparing for the Exam**

Certification preparation learning guides for the new version of the Terraform Associate will be posted soon. For now, use the current [Terraform Associate Tutorial List](https://learn.hashicorp.com/collections/terraform/certification-associate-tutorials) to start your studying.

**Exam Details**

|  |  |
| --- | --- |
| **Assessment Type** | Multiple choice |
| **Format** | Online proctored |
| **Duration** | 1 hour |
| **Price** | $70.50 USD plus locally applicable taxes and fees Free retake **not included** |
| **Language** | English |
| **Expiration** | 2 years |

# Exam Objectives

## **Understand infrastructure as code (IaC) concepts**

### Explain what IaC is

### Describe advantages of IaC patterns

## **Understand the purpose of Terraform (vs other IaC)**

### Explain multi-cloud and provider-agnostic benefits

### Explain the benefits of state

## **Understand Terraform basics**

### Install and version Terraform providers

### Describe plugin-based architecture

### Write Terraform configuration using multiple providers

### Describe how Terraform finds and fetches providers

## **Use Terraform outside of core workflow**

### Describe when to use terraform import to import existing infrastructure into your Terraform state

### Use terraform state to view Terraform state

### Describe when to enable verbose logging and what the outcome/value is

## **Interact with Terraform modules**

### Contrast and use different module source options including the public Terraform Module Registry

### Interact with module inputs and outputs

### Describe variable scope within modules/child modules

### Set module version

## **Use the core Terraform workflow**

### Describe Terraform workflow ( Write -> Plan -> Create )

### Initialize a Terraform working directory (terraform init)

### Validate a Terraform configuration (terraform validate)

### Generate and review an execution plan for Terraform (terraform plan)

### Execute changes to infrastructure with Terraform (terraform apply)

### Destroy Terraform managed infrastructure (terraform destroy)

### Apply formatting and style adjustments to a configuration (terraform fmt)

## **Implement and maintain state**

### Describe default local backend

### Describe state locking

### Handle backend and cloud integration authentication methods

### Differentiate remote state back end options

### Manage resource drift and Terraform state

### Describe backend block and cloud integration in configuration

### Understand secret management in state files

## **Read, generate, and modify configuration**

### Demonstrate use of variables and outputs

### Describe secure secret injection best practice

### Understand the use of collection and structural types

### Create and differentiate resource and data configuration

### Use resource addressing and resource parameters to connect resources together

### Use HCL and Terraform functions to write configuration

### Describe built-in dependency management (order of execution based)

## **Understand Terraform Cloud capabilities**

## Explain how Terraform Cloud helps to manage infrastructure

## Describe how Terraform Cloud enables collaboration and governance