



HashiCorp Terraform Associate

**HashiCorp Infrastructure Automation
Certification Questions & Answers**

Get Instant Access to Vital
Exam Acing Materials |
Study Guide | Sample
Questions | Practice Test

TERRAFORM ASSOCIATE

[HashiCorp Certified - Terraform Associate](#)

57 Questions Exam – 70% Cut Score – Duration of 60 minutes



Table of Contents:

Discover More about the Terraform Associate Certification	2
HashiCorp Terraform Associate Certification Details: ...	2
Terraform Associate Syllabus:.....	3
Broaden Your Knowledge with HashiCorp Terraform Associate Sample Questions:.....	4
Avail the Study Guide to Pass HashiCorp Terraform Associate Exam:.....	7
Career Benefits:	8

Discover More about the Terraform Associate Certification

Are you interested in passing the HashiCorp Terraform Associate exam? First discover, who benefits from the Terraform Associate certification. The Terraform Associate is suitable for a candidate if he wants to learn about Infrastructure Automation. Passing the Terraform Associate exam earns you the HashiCorp Certified - Terraform Associate title.

While preparing for the Terraform Associate exam, many candidates struggle to get the necessary materials. But do not worry; your struggling days are over. The Terraform Associate PDF contains some of the most valuable preparation tips and the details and instant access to useful Terraform Associate study materials [just at one click](#).

HashiCorp Terraform Associate Certification Details:

Exam Name	HashiCorp Certified Terraform Associate (Infrastructure Automation)
Exam Code	Infrastructure Automation
Exam Price	\$70 USD
Duration	60 minutes
Number of Questions	57
Passing Score	Pass / Fail (Approx 70%)
Recommended Training / Books	Prepare for the exam
Schedule Exam	Cloud Engineer Certification Exam Portal
Sample Questions	HashiCorp Terraform Associate Sample Questions
Recommended Practice	HashiCorp Certified - Terraform Associate Practice Test

Terraform Associate Syllabus:

Section	Objectives
Understand infrastructure as code (IaC) concepts	<ul style="list-style-type: none"> - Explain what IaC is - Describe advantages of IaC patterns
Understand the purpose of Terraform (vs other IaC)	<ul style="list-style-type: none"> - Explain multi-cloud and provider-agnostic benefits - Explain the benefits of state
Understand Terraform basics	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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

Section	Objectives
	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- Explain how Terraform Cloud helps to manage infrastructure- Describe how Terraform Cloud enables collaboration and governance

Broaden Your Knowledge with HashiCorp Terraform Associate Sample Questions:

Question: 1

During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

- a) Terraform attempts to provision the resource up to three times before exiting with an error
- b) it is automatically deleted
- c) the terraform plan is rolled back and all provisioned resources are removed
- d) the resource is marked as tainted

Answer: d

Question: 2

You have defined the values for your variables in the file terraform.tfvars, and saved it in the same directory as your Terraform configuration.

Which of the following commands will use those values when creating an execution plan?

- a) terraform plan
- b) terraform plan -var-file=terraform.tfvars
- c) All of the above
- d) None of the above

Answer: c

Question: 3

What happens when you apply Terraform configuration?

Choose TWO correct answers.

- a) Terraform makes any infrastructure changes defined in your configuration.
- b) Terraform gets the plugins that the configuration requires.
- c) Terraform updates the state file with any configuration changes it made.
- d) Terraform corrects formatting errors in your configuration.
- e) Terraform destroys and recreates all your infrastructure from scratch.

Answer: a, c

Question: 4

A provider alias is used for what purpose in a Terraform configuration file?

- a) alias isn't used with providers, they are used with provisioners
- b) to signify what cloud-based region to deploy resources
- c) to use as shorthand for resources to be deployed with the referenced provider
- d) using the same provider with different configurations for different resources

Answer: d

Question: 5

Consider the following Terraform 0.12 configuration snippet:

```
variable "vpc_cidrs" {
  type = map
  default = {
    us-east-1 = "10.0.0.0/16"
    us-east-2 = "10.1.0.0/16"
    us-west-1 = "10.2.0.0/16"
    us-west-2 = "10.3.0.0/16"
  }
}

resource "aws_vpc" "shared" {
  cidr_block = _____
}
```

How would you define the `cidr_block` for `us-east-1` in the `aws_vpc` resource using a variable?

- a) `var.vpc_cidrs["us-east-1"]`
- b) `var.vpc_cidrs.0`
- c) `vpc_cidrs["us-east-1"]`
- d) `var.vpc_cidrs[0]`

Answer: a

Question: 6

When running a terraform plan, how can you save the plan so it can be applied at a later time?

- a) you cannot save a plan
- b) use the -file parameter
- c) use the -save parameter
- d) use the -out parameter

Answer: d

Question: 7

Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments?

Choose TWO correct answers.

- a) terraform plan
- b) terraform state
- c) terraform apply
- d) terraform validate
- e) terraform output

Answer: a, c

Question: 8

Published modules via the Terraform Registry provide which of the following benefits?

(select four)

- a) support versioning
- b) automatically generated documentation
- c) allow browsing version histories
- d) support from any code repo
- e) show examples and READMEs

Answer: a, b, c, e

Question: 9

Which of the following represents a feature of Terraform Cloud that is NOT free to customers?

- a) VCS integration
- b) team management and governance
- c) private module registry
- d) workspace management

Answer: b

Question: 10

Infrastructure as Code (IaC) provides many benefits to help organizations deploy application infrastructure much faster than clicking around in the console. What are additional benefits to IaC?

(select three)

- a) allows infrastructure to be versioned
- b) creates a blueprint of your data center
- c) code can easily be shared and reused
- d) can always be used to deploy the latest features and services
- e) eliminates parallelism

Answer: a, b, c

Avail the Study Guide to Pass HashiCorp Terraform Associate Exam:

- Find out about the Terraform Associate syllabus topics. Visiting the official site offers an idea about the exam structure and other important study resources. Going through the syllabus topics help to plan the exam in an organized manner.
- Once you are done exploring the [Terraform Associate syllabus](#), it is time to plan for studying and covering the syllabus topics from the core. Chalk out the best plan for yourself to cover each part of the syllabus in a hassle-free manner.
- A study schedule helps you to stay calm throughout your exam preparation. It should contain your materials and thoughts like study hours, number of topics for daily studying mentioned on it. The best bet to clear the exam is to follow your schedule rigorously.
- The candidate should not miss out on the scope to learn from the Terraform Associate training. Joining the HashiCorp provided training for Terraform Associate exam helps a candidate to strengthen his practical knowledge base from the certification.
- Learning about the probable questions and gaining knowledge regarding the exam structure helps a lot. Go through the [Terraform Associate sample questions](#) and boost your knowledge
- Make yourself a pro through online practicing the syllabus topics. Terraform Associate practice tests would guide you on your strengths and weaknesses regarding the syllabus topics. Through rigorous

practicing, you can improve the weaker sections too. Learn well about time management during exam and become confident gradually with practice tests.

Career Benefits:

- Passing the Terraform Associate exam, helps a candidate to prosper highly in his career. Having the certification on the resume adds to the candidate's benefit and helps to get the best opportunities.

Here Is the Trusted Practice Test for the Terraform Associate Certification

VMExam.Com is here with all the necessary details regarding the Terraform Associate exam. We provide authentic practice tests for the Terraform Associate exam. What do you gain from these practice tests? You get to experience the real exam-like questions made by industry experts and get a scope to improve your performance in the actual exam. Rely on VMExam.Com for rigorous, unlimited two-month attempts on the [Terraform Associate practice tests](#), and gradually build your confidence. Rigorous practice made many aspirants successful and made their journey easy towards grabbing the HashiCorp Certified - Terraform Associate.

Start Online practice of Terraform Associate Exam by visiting URL
<https://www.vmexam.com/hashicorp/hashicorp-certified-terraform-associate>