vmexam.com
#1 Cloud and VM Certification Guide

HashiCorp Terraform Authoring and Operations

HASHICORP TERRAFORM AUTHORING AND OPERATIONS CERTIFICATION QUESTIONS & ANSWERS

Exam Summary – Syllabus – Questions

TERRAFORM AUTHORING AND OPERATIONS

HashiCorp Certified - Terraform Authoring and Operations Professional

57 Questions Exam – 70% Cut Score – Duration of 240 minutes

www.VMExam.com

Table of Contents

Know Your Terraform Authoring and Operations Certification Well:

The Terraform Authoring and Operations is best suitable for candidates who want to gain knowledge in the HashiCorp Infrastructure Automation. Before you start your Terraform Authoring and Operations preparation you may struggle to get all the crucial Terraform Authoring and Operations materials like Terraform Authoring and Operations syllabus, sample questions, study guide.

But don't worry the Terraform Authoring and Operations PDF is here to help you prepare in a stress free manner.

The PDF is a combination of all your queries like-

- What is in the Terraform Authoring and Operations syllabus?
- How many questions are there in the Terraform Authoring and Operations exam?
- Which Practice test would help me to pass the Terraform Authoring and Operations exam at the first attempt?

Passing the Terraform Authoring and Operations exam makes you HashiCorp Certified - Terraform Authoring and Operations Professional. Having the Terraform Authoring and Operations certification opens multiple opportunities for you. You can grab a new job, get a higher salary or simply get recognition within your current organization.

HashiCorp Terraform Authoring and Operations Certification Details:

Exam Name	HashiCorp Certified Terraform Authoring and Operations Professional
Exam Code	Terraform Authoring and Operations
Exam Price	$295 USD
Duration	240 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 Authoring and Operations Sample Questions**
Recommended Practice	**HashiCorp Certified - Terraform Authoring and Operations Professional Practice Test**

Terraform Authoring and Operations Syllabus:

Section	Objectives
Manage resource lifecycle	- Initialize a configuration using terraform init and its options - Generate an execution plan using terraform plan and its options - Apply configuration changes using terraform apply and its options - Destroy resources using terraform destroy and its options - Manage resource state, including importing resources and reconciling resource drift
Develop and troubleshoot dynamic configuration	- Use language features to validate configuration - Query providers using data sources - Compute and interpolate data using HCL functions - Use meta-arguments in configuration - Configure input variables and outputs, including complex types - Analyze best practices for managing sensitive data, such as using Vault for secrets management
Develop collaborative Terraform workflows	- Manage the Terraform binary, providers, and modules using version constraints - Configure remote state - Use the Terraform workflow in automation - Share data across configurations and workspaces
Create, maintain, and use Terraform modules	- Create a module - Use a module in configuration - Refactor a module and use module versioning - Refactor an existing configuration into modules

Section	Objectives
Configure and use Terraform providers	- Understand Terraform's plugin-based architecture - Configure providers, including aliasing, versioning, sourcing, and managing upgrades - Manage provider authentication - Troubleshoot provider errors
Collaborate on infrastructure as code using HCP Terraform	- Analyze the HCP Terraform run workflow - Understand HCP Terraform workspaces and their configuration options, including access management - Manage provider credentials in HCP Terraform - Analyze policy as code and governance features

HashiCorp Terraform Authoring and Operations Sample Questions:

Question: 1

What are two benefits of using HCP Terraform’s policy-as-code features?

(Choose two)

a) Simplifies provider authentication processes
b) Enforces organizational policies before execution
c) Provides detailed cost analysis of resources
d) Prevents non-compliant infrastructure changes

Answer: b, d

Question: 2

Which two features of HCP Terraform help enforce governance and compliance?

(Choose two)

a) Module versioning
b) Sentinel policies
c) Remote execution
d) Role-based access control

Answer: b, d

Question: 3

What practices ensure consistent and reliable module usage in collaborative workflows?

(Choose two)

a) Using module version constraints
b) Creating a local copy of modules for each user
c) Storing modules in the Terraform public registry
d) Hosting private modules in a private registry

Answer: a, d

Question: 4

Which statement best describes how HCP Terraform handles provider credentials?

a) Credentials must be hardcoded into Terraform configuration files
b) Credentials can be securely stored and managed in HCP Terraform
c) HCP Terraform does not manage provider credentials
d) Provider credentials are shared across all workspaces

Answer: b

Question: 5

Which of the following meta-arguments can be used to dynamically configure multiple instances of a provider?

a) for_each
b) count
c) depends_on
d) lifecycle

Answer: a

Question: 6

What is the best way to securely inject sensitive data like passwords into a Terraform configuration?

a) Use Vault or similar secret management tools.
b) Store them in a .tf file and version control it.
c) Pass them directly in the terraform plan command.
d) Store them in the terraform.tfvars file.

Answer: a

Question: 7

Which Terraform feature is used to validate the correctness of a configuration without interacting with any providers?

a) terraform init
b) terraform plan
c) terraform validate
d) terraform fmt

Answer: c

Question: 8

When creating a module, which file is considered the entry point for the module's configuration?

a) variables.tf
b) output.tf
c) main.tf
d) provider.tf

Answer: c

Question: 9

How can you call a module stored in a private Git repository?

a) module "example" { source = "git://private-repo.git" }
b) module "example" { source = "git::https://username@private-repo.git" }
c) module "example" { source = "https://private-repo.git" }
d) module "example" { path = "./private-repo" }

Answer: b

Question: 10

What command would you use to upgrade a provider to its latest version?

a) terraform validate
b) terraform apply -upgrade
c) terraform init -upgrade
d) terraform get -update

Answer: c

Study Guide to Crack HashiCorp Terraform Authoring and Operations Exam:

- Getting details of the Terraform Authoring and Operations syllabus, is the first step of a study plan. This pdf is going to be of ultimate help. Completion of the syllabus is must to pass the Terraform Authoring and Operations exam.
- Making a schedule is vital. A structured method of preparation leads to success. A candidate must plan his schedule and follow it rigorously to attain success.
- Joining the HashiCorp provided training for Terraform Authoring and Operations exam could be of much help. If there is specific training for the exam, you can discover it from the link above.
- Read from the Terraform Authoring and Operations sample questions to gain your idea about the actual exam questions. In this PDF useful sample questions are provided to make your exam preparation easy.
- Practicing on Terraform Authoring and Operations practice tests is must. Continuous practice will make you an expert in all syllabus areas.