HashiCorp Terraform Associate (003) Exam Quiz

I've created a comprehensive quiz to help you prepare for the HashiCorp Terraform Associate (003) certification exam. This quiz covers all 9 exam objectives and includes various question types that you'll encounter on the actual exam.

Exam Overview

The HashiCorp Terraform Associate (003) certification exam validates your knowledge of Infrastructure as Code (IaC) concepts and practical Terraform skills[1][2]. Here are the key details:

- **Duration**: 60 minutes
- **Questions**: 57 questions
- **Format**: Multiple choice, multiple select, true/false, and fill-in-the-blank
- **Passing Score**: Approximately 70%
- **Cost**: $70.50 USD plus taxes
- **Validity**: 2 years
- **Format**: Online proctored exam

Practice Quiz Questions

Domain 1: Understand Infrastructure as Code (IaC) Concepts

Question 1: What is Infrastructure as Code (IaC)?

- A. A method of managing applications using code
- B. A practice of managing and provisioning infrastructure through machine-readable definition files
- C. A way to write application code for cloud services
- D. A database management technique

Answer: B. A practice of managing and provisioning infrastructure through machine-readable definition files[3]

Question 2: Which of the following are advantages of IaC? (Select all that apply)

- A. Version control
- B. Repeatability
- C. Consistency

- D. Manual configuration

Answer: A, B, C. Version control, repeatability, and consistency are key advantages of IaC[3]

Domain 2: Understand Terraform's Purpose (vs other IaC)

Question 3: What is a key benefit of Terraform's multi-cloud approach?

- A. It only works with AWS
- B. It provides vendor lock-in
- C. It allows you to manage resources across multiple cloud providers using the same workflow
- D. It requires separate tools for each cloud

Answer: C. It allows you to manage resources across multiple cloud providers using the same workflow[3]

Question 4: True or False: Terraform state helps track the real-world resources that correspond to your configuration.

Answer: True. The state file is essential for Terraform to map configuration to real-world resources[3]

Domain 3: Understand Terraform Basics

Question 5: Which command initializes a new Terraform working directory?

- A. terraform plan
- B. terraform apply
- C. terraform init
- D. terraform validate

Answer: C. terraform init[4][5]

Question 6: What file extension is commonly used for Terraform configuration files?

- A. .yaml
- B. .json
- C. .tf
- D. .hcl

Answer: C. .tf[4]

Question 7: Fill in the blank: Terraform uses ____ (HashiCorp Configuration Language) for writing configuration files.

Answer: HCL[6]

Domain 4: Use Terraform CLI (outside of core workflow)

Question 8: Which command would you use to import existing infrastructure into Terraform state?

- A. terraform get
- B. terraform import
- C. terraform refresh
- D. terraform sync

Answer: B. terraform import[7][3]

Question 9: What does the `terraform state list` command do?

- A. Creates a new state file
- B. Shows all resources in the Terraform state
- C. Deletes the state file
- D. Validates the state file

Answer: B. Shows all resources in the Terraform state[3]

Domain 5: Interact with Terraform Modules

Question 10: What is a Terraform module?

- A. A single resource definition
- B. A collection of Terraform configuration files that together define a set of related resources
- C. A cloud provider plugin
- D. A state management tool

Answer: B. A collection of Terraform configuration files that together define a set of related resources[4][8]

Question 11: Where can you source Terraform modules from? (Select all that apply)

- A. Local filesystem
- B. Terraform Registry
- C. Git repositories
- D. HTTP URLs

Answer: A, B, C, D. Terraform supports multiple module sources[3]

Domain 6: Navigate Terraform Workflow

Question 12: What is the correct order of the core Terraform workflow?

- A. Apply → Plan → Write
- B. Write → Apply → Plan
- C. Write → Plan → Apply
- D. Plan → Write → Apply

Answer: C. Write → Plan → Apply[3]

Question 13: Which command generates and shows an execution plan?

- A. terraform apply
- B. terraform plan
- C. terraform show
- D. terraform output

Answer: B. terraform plan[9][3]

Question 14: What does `terraform destroy` do?

- A. Deletes Terraform configuration files
- B. Removes the state file
- C. Destroys all resources defined in the Terraform configuration
- D. Uninstalls Terraform

Answer: C. Destroys all resources defined in the Terraform configuration[9]

Domain 7: Implement and Maintain State

Question 15: True or False: Terraform state files should be stored in version control.

Answer: False. State files contain sensitive information and should not be stored in version control[3]

Question 16: What is state locking in Terraform?

- A. A feature that prevents multiple users from simultaneously modifying the same state
- B. A way to encrypt state files
- C. A method to backup state files
- D. A process to validate state files

Answer: A. A feature that prevents multiple users from simultaneously modifying the same state[3]

Question 17: Which backend type stores state locally?

- A. remote

- B. s3
- C. local
- D. http

Answer: C. local[3]

Domain 8: Read, Generate, and Modify Configuration

Question 18: How do you reference a variable named "instance_type" in Terraform?

- A. instance_type
- B. ${instance_type}
- C. var.instance_type
- D. variable.instance_type

Answer: C. var.instance_type[10]

Question 19: Which block type is used to define input variables?

- A. resource
- B. data
- C. variable
- D. output

Answer: C. variable[8]

Question 20: True or False: Data sources in Terraform are used to create new infrastructure resources.

Answer: False. Data sources are used to fetch information about existing infrastructure[6]

Domain 9: Understand Terraform Cloud and Enterprise Capabilities

Question 21: What is a key feature of Terraform Cloud?

- A. Local execution only
- B. Remote state management and collaboration features
- C. Only works with AWS
- D. Requires on-premises installation

Answer: B. Remote state management and collaboration features[3]

Question 22: Which of the following are Terraform Cloud features? (Select all that apply)

- A. Remote execution
- B. Team collaboration
- C. Cost estimation

- D. Policy enforcement

Answer: A, B, C, D. All are features of Terraform Cloud[3]

Advanced Practice Questions

Question 23: In the following code snippet, how would you reference the "us-east-1" CIDR block?

```
variable "vpc_cidrs" {
  type = map(string)
  default = {
    us-east-1 = "10.0.0.0/16"
    us-east-2 = "10.1.0.0/16"
  }
}
```

- A. var.vpc_cidrs.us-east-1
- B. var.vpc_cidrs["us-east-1"]
- C. vpc_cidrs.us-east-1
- D. var.vpc_cidrs

Answer: B. var.vpc_cidrs["us-east-1"][10]

Question 24: What happens when you run `terraform plan` with no changes to make?

- A. Terraform throws an error
- B. Terraform shows "No changes. Infrastructure is up-to-date."
- C. Terraform still applies changes
- D. Terraform destroys resources

Answer: B. Terraform shows "No changes. Infrastructure is up-to-date."[3]

Question 25: True or False: Sensitive variables in Terraform are automatically hidden in the CLI output.

Answer: True. Terraform marks sensitive values and hides them from CLI output[3]

Study Tips and Resources

Key Study Areas

1. **Infrastructure as Code concepts** - Understand the benefits and use cases[2]
2. **Terraform CLI commands** - Practice with init, plan, apply, destroy, import, and state commands[3]
3. **Configuration syntax** - Learn HCL syntax, variables, outputs, and data sources[3]
4. **State management** - Understand local vs remote backends, state locking[3]

5. **Modules** - Know how to create, use, and version modules[3]
6. **Terraform Cloud features** - Remote execution, collaboration, governance[3]

Recommended Study Resources

- **Official HashiCorp Learn tutorials**[2]
- **Practice exams** by Bryan Krausen on Udemy[11][12]
- **Free practice questions** from K21Academy[8] and Whizlabs[5]
- **Official documentation** on developer.hashicorp.com[3]
- **Hands-on labs** - Set up a free cloud account and practice[13]

Exam Strategy

- **Read questions carefully** - The exam isn't designed to trick you[10]
- **Manage your time** - 60 minutes for 57 questions (about 1 minute per question)[1]
- **Use elimination** - Rule out obviously wrong answers first[12]
- **Flag difficult questions** - Return to them if time permits[12]
- **Don't leave blanks** - There's no penalty for wrong answers[1]

This quiz covers the essential topics you'll encounter on the HashiCorp Terraform Associate (003) exam. Practice regularly, get hands-on experience with Terraform, and review the official documentation to ensure you're well-prepared for certification success[13][12]!