TERRAFORM-ASSOCIATE-003 Q&As

HashiCorp Certified: Terraform Associate (003)

Pass HashiCorp TERRAFORM-ASSOCIATE-003 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/terraform-associate-003.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by HashiCorp Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

FILL IN THE BLANK

What is the name of the default file where Terraform stores the state?

Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

Correct Answer: Terraform.tfstate

Terraform.tfstate

The name of the default file where Terraform stores the state is terraform.tfstate. This file contains a JSON representation of the current state of the infrastructure managed by Terraform. Terraform uses this file to track the metadata and attributes of the resources, and to plan and apply changes. By default, Terraform stores the state file locally in the same directory as the configuration files, but it can also be configured to store the state remotely in a backend. References = [Terraform State], [State File Format]

QUESTION 2

You have created a main.tf Terraform configuration consisting of an application server, a database and a load balanced. You ran terraform apply and Terraform created all of the resources successfully. Now you realize that you do not actually need the load balancer, so you run terraform destroy without any flags. What will happen?

A. Terraform will prompt you to pick which resource you want to destroy

B. Terraform will destroy the application server because it is listed first in the code

C. Terraform will prompt you to confirm that you want to destroy all the infrastructure

D. Terraform will destroy the main, tf file

E. Terraform will immediately destroy all the infrastructure

Correct Answer: C

This is what will happen if you run terraform destroy without any flags, as it will attempt to delete all the resources that are associated with your current working directory or workspace. You can use the -target flag to specify a particular resource that you want to destroy.

QUESTION 3

What does the default "local" Terraform backend store?

A. tfplan files

B. State file

C. Provider plugins

D. Terraform binary

Correct Answer: B

The default "local" Terraform backend stores the state file in a local file named terraform.tfstate, which can be used to track and manage the state of your infrastructure3.

QUESTION 4

How could you reference an attribute from the vsphere_datacenter data source for use with the datacenter_id argument within the vsphere_folder resource in the following configuration?

```
data "vsphere_datacenter" "dc" {}

resource "vsphere_folder" "parent" {
          path = "Production"
          type = "vm"
  datacenter_id = _____________________
}
```

A. Data.vsphere_datacenter.DC.id

B. Vsphere_datacenter.dc.id

C. Data,dc,id

D. Data.vsphere_datacenter,dc

Correct Answer: A

The correct way to reference an attribute from the vsphere_datacenter data source for use with the datacenter_id argument within the vsphere_folder resource in the following configuration is data.vsphere_datacenter.dc.id. This follows the

syntax for accessing data source attributes, which is data.TYPE.NAME.ATTRIBUTE. In this case, the data source type is vsphere_datacenter, the data source name is dc, and the attribute we want to access is id. The other options are

incorrect because they either use the wrong syntax, the wrong punctuation, or the wrong case. References = [Data Source:

vsphere_datacenter], [Data Source: vsphere_folder], [Expressions: Data Source References]

QUESTION 5

Which of these commands makes your code more human readable?

A. Terraform validate

B. Terraform output

C. Terraform show

D. Terraform fmt

Correct Answer: D

The command that makes your code more human readable is terraform fmt. This command is used to rewrite Terraform configuration files to a canonical format and style, following the Terraform language style conventions and other minor adjustments for readability. The command is optional, opinionated, and has no customization options, but it is recommended to ensure consistency of style across different Terraform codebases. Consistency can help your team understand the code more quickly and easily, making the use of terraform fmt very important. You can run this command on your configuration files before committing them to source control or as part of your CI/CD pipeline. References = : Command: fmt : Using Terraform fmt Command to Format Your Terraform Code