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

When should you write Terraform configuration files for existing infrastructure that you want to start managing with
Terraform?

A. You can import infrastructure without corresponding Terraform code

B. Terraform will generate the corresponding configuration files for you

C. Before you run terraform Import

D. After you run terraform import

Correct Answer: C

You need to write Terraform configuration files for the existing infrastructure that you want to import into Terraform,
otherwise Terraform will not know how to manage it. The configuration files should match the type and name of the
resources that you want to import.

QUESTION 2

You are using a networking module in your Terraform configuration with the name label my-network. In your main
configuration you have the following code:

```
output "net_id" {
  value = module.my_network.vnet_id
}
```

When you run terraform validate, you get the following error:

```
Error: Reference to undeclared output value

  on main.tf line 12, in output "net_id":
  12:   value = module.my_network.vnet_id
```

What must you do to successfully retrieve this value from your networking module?

A. Change the reference value to my-network,outputs,vmet_id

B. Define the attribute vmet_id as a variable in the networking modeule

C. Define the attribute vnet_id as an output in the networking module

D. Change the reference value module.my,network,outputs,vnet_id

Correct Answer: C

This is what you must do to successfully retrieve this value from your networking module, as it will expose the attribute as an output value that can be referenced by other modules or resources. The error message indicates that the networking module does not have an output value named vnet_id, which causes the reference to fail.

QUESTION 3

How does the Terraform cloud integration differ from other state backends such as S3, Consul,etc?

A. It can execute Terraform runs on dedicated infrastructure in Terraform Cloud

B. It doesn\\'t show the output of a terraform apply locally

C. It is only arable lo paying customers

D. All of the above

Correct Answer: A

This is how the Terraform Cloud integration differs from other state backends such as S3, Consul, etc., as it allows you to perform remote operations on Terraform Cloud\\'s servers instead of your local machine. The other options are either incorrect or irrelevant.

QUESTION 4

When using Terraform to deploy resources into Azure, which scenarios are true regarding state files? (Choose two.)

A. When you change a Terraform-managed resource via the Azure Cloud Console, Terraform updates the state file to reflect the change during the next plan or apply

B. Changing resources via the Azure Cloud Console records the change in the current state file

C. When you change a resource via the Azure Cloud Console, Terraform records the changes in a new state file

D. Changing resources via the Azure Cloud Console does not update current state file

Correct Answer: AD

Terraform state is a representation of the infrastructure that Terraform manages. Terraform uses state to track the current status of the resources it creates and to plan future changes. However, Terraform state is not aware of any changes made to the resources outside of Terraform, such as through the Azure Cloud Console, the Azure CLI, or the Azure API. Therefore, changing resources via the Azure Cloud Console does not update the current state file, and it may cause inconsistencies or conflicts with Terraform\\'s desired configuration. To avoid this, it is recommended to manage resources exclusively through Terraform or to use the terraform import command to bring existing resources under Terraform\\'s control. When you change a Terraform-managed resource via the Azure Cloud Console, Terraform does not immediately update the state file to reflect the change. However, the next time you run terraform plan or terraform apply, Terraform will compare the state file with the actual state of the resources in Azure and detect any drifts or differences. Terraform will then update the state file to match the current state of the resources and show you the proposed changes in the execution plan. Depending on the configuration and the change, Terraform may try to undo the change, modify the resource further, or recreate the resource entirely. To avoid unexpected or destructive changes, it is recommended to review the execution plan carefully before applying it or to use the terraform refresh command to update the state file without applying any changes. References = Purpose of Terraform State, Terraform State, Managing State, Importing Infrastructure, [Command: plan], [Command: apply], [Command: refresh]

QUESTION 5

Which option cannot be used to keep secrets out of Terraform configuration files?

A. A Terraform provider

B. Environment variables

C. A -var flag

D. secure string

Correct Answer: D

A secure string is not a valid option to keep secrets out of Terraform configuration files. A secure string is a feature of AWS Systems Manager Parameter Store that allows you to store sensitive data encrypted with a KMS key. However, Terraform does not support secure strings natively and requires a custom data source to retrieve them. The other options are valid ways to keep secrets out of Terraform configuration files. A Terraform provider can expose secrets as data sources that can be referenced in the configuration. Environment variables can be used to set values for input variables that contain secrets. A -var flag can be used to pass values for input variables that contain secrets from the command line or a file. References = [AWS Systems Manager Parameter Store], [Terraform AWS Provider Issue #55], [Terraform Providers], [Terraform Input Variables]

Latest TERRAFORM-
ASSOCIATE-003 Dumps

TERRAFORM-
ASSOCIATE-003 PDF
Dumps

TERRAFORM-
ASSOCIATE-003 Practice
Test