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.certbus.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

A provider configuration block is required in every Terraform configuration.

Example:

```
provider "provider_name" {
  ...
}
```

A. True

B. False

Correct Answer: B

A provider configuration block is not required in every Terraform configuration. A provider configuration block can be omitted if its contents would otherwise be empty. Terraform assumes an empty default configuration for any provider that is not explicitly configured. However, some providers may require some configuration arguments (such as endpoint URLs or cloud regions) before they can be used. A provider\\'s documentation should list which configuration arguments it expects. For providers distributed on the Terraform Registry, versioned documentation is available on each provider\\'s page, via the "Documentation" link in the provider\\'s header1. References = [Provider Configuration]1

QUESTION 2

How is terraform import run?

A. As a part of terraform init

B. As a part of terraform plan

C. As a part of terraform refresh

D. By an explicit call

E. All of the above

Correct Answer: D

The terraform import command is not part of any other Terraform workflow. It must be explicitly invoked by the user with the appropriate arguments, such as the resource address and the ID of the existing infrastructure to import. References = [Importing Infrastructure]

QUESTION 3

How can you trigger a run in a Terraform Cloud workspace that is connected to a Version Control System (VCS) repository?

A. Only Terraform Cloud organization owners can set workspace variables on VCS connected workspaces

B. Commit a change to the VCS working directory and branch that the Terraform Cloud workspace is connected to

C. Only Terraform Cloud organization owners can approve plans in VCS connected workspaces

D. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces

Correct Answer: B

This will trigger a run in the Terraform Cloud workspace, which will perform a plan and apply operation on the infrastructure defined by the Terraform configuration files in the VCS repository.

QUESTION 4

You can configure Terraform to log to a file using the TF_LOG environment variable.

A. True

B. False

Correct Answer: A

You can configure Terraform to log to a file using the TF_LOG environment variable. This variable can be set to one of the log levels: TRACE, DEBUG, INFO, WARN or ERROR. You can also use the TF_LOG_PATH environment variable to specify a custom log file location. References = : Debugging Terraform

QUESTION 5

Module variable assignments are inherited from the parent module and you do not need to explicitly set them.

A. True

B. False

Correct Answer: B

Module variable assignments are not inherited from the parent module and you need to explicitly set them using the source argument. This allows you to customize the behavior of each module instance.

QUESTION 6

You\\'ve used Terraform to deploy a virtual machine and a database. You want to replace this virtual machine instance with an identical one without affecting the database. What is the best way to achieve this using Terraform?

A. Use the terraform state rm command to remove the VM from state file

B. Use the terraform taint command targeting the VMs then run terraform plan and terraform apply

C. Use the terraform apply command targeting the VM resources only

D. Delete the Terraform VM resources from your Terraform code then run terraform plan and terraform apply

Correct Answer: B

The terraform taint command marks a resource as tainted, which means it will be destroyed and recreated on the next apply. This way, you can replace the VM instance without affecting the database or other resources. References = [Terraform Taint]

QUESTION 7

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 8

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]

QUESTION 9

You have never used Terraform before and would like to test it out using a shared team account for a cloud provider. The shared team account already contains 15 virtual machines (VM). You develop a Terraform configuration containing one VM. perform terraform apply, and see that your VM was created successfully. What should you do to delete the newly-created VM with Terraform?

A. The Terraform state file contains all 16 VMs in the team account. Execute terraform destroy and select the newly-created VM.

B. Delete the Terraform state file and execute terraform apply.

C. The Terraform state file only contains the one new VM. Execute terraform destroy.

D. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.

Correct Answer: C

This is the best way to delete the newly-created VM with Terraform, as it will only affect the resource that was created by your configuration and state file. The other options are either incorrect or inefficient.

QUESTION 10

Which two steps are required to provision new infrastructure in the Terraform workflow? Choose two correct answers.

A. Plan

B. Import

C. Alidate

D. Init

E. apply

Correct Answer: DE

The two steps that are required to provision new infrastructure in the Terraform workflow are init and apply. The terraform init command initializes a working directory containing Terraform configuration files. It downloads and installs the provider plugins that are needed for the configuration, and prepares the backend for storing the state. The terraform apply command applies the changes required to reach the desired state of the configuration, as described by the resource definitions in the configuration files. It shows a plan of the proposed changes and asks for confirmation before making any changes to the infrastructure. References = [The Core Terraform Workflow], [Initialize a Terraform working directory with init], [Apply Terraform Configuration with apply]

QUESTION 11

How does Terraform manage most dependencies between resources?

A. Terraform will automatically manage most resource dependencies

B. Using the depends_on parameter

C. By defining dependencies as modules and including them in a particular order

D. The order that resources appear in Terraform configuration indicates dependencies

Correct Answer: A

This is how Terraform manages most dependencies between resources, by using the references between them in the configuration files. For example, if resource A depends on resource B, Terraform will create resource B first and then pass its attributes to resource A.

QUESTION 12

You cannot install third party plugins using terraform init.

A. True

B. False

Correct Answer: B

You can install third party plugins using terraform init, as long as you specify the plugin directory in your configuration or as a command-line argument. You can also use the terraform providers mirror command to create a local mirror of providers from any source.

QUESTION 13

Where does the Terraform local backend store its state?

A. In the terraform file

B. In the /tmp directory

C. In the terraform,tfstate file

D. In the user\\'s terraform,state file

Correct Answer: C

This is where the Terraform local backend stores its state, by default, unless you specify a different file name or location in your configuration. The local backend is the simplest backend type that stores the state file on your local disk.

QUESTION 14

You are writing a child Terraform module that provisions an AWS instance. You want to reference the IP address returned by the child module in the root configuration. You name the instance resource "main\\'. Which of these is the correct way to define the output value?

A.
```
output "instance_ip_addr" {
  return aws_instance.main.private_ip
}
```

B.
```
output "aws_instance.instance_ip_addr" {
  return aws_instance.main.private_ip
}
```

C.
```
output "aws_instance.instance_ip_addr" {
  value = ${main.private_ip}
}
```

D.
```
output "instance_ip_addr" {
  value = aws_instance.main.private_ip
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

QUESTION 15

One remote backend configuration always maps to a single remote workspace.

A. True

B. False

Correct Answer: A

The remote backend can work with either a single remote Terraform Cloud workspace, or with multiple similarly-named remote workspaces (like networking-dev and networking-prod). The workspaces block of the backend configuration determines which mode it uses. To use a single remote Terraform Cloud workspace, set workspaces.name to the remote workspace\\'s full name (like networking-prod). To use multiple remote workspaces, set workspaces.prefix to a prefix used in all of the desired remote workspace names. For example, set prefix = "networking-" to use Terraform cloud workspaces with names like networking-dev and networking-prod. This is helpful when mapping multiple Terraform CLI workspaces used in a single Terraform configuration to multiple Terraform Cloud workspaces3. However, one remote backend configuration always maps to a single remote workspace, either by name or by prefix. You cannot use both name and prefix in the same backend configuration, or omit both. Doing so will result in a configuration error3. References = [Backend Type: remote]3

TERRAFORM-ASSOCIATE-003 PDF Dumps

TERRAFORM-ASSOCIATE-003 VCE Dumps

TERRAFORM-ASSOCIATE-003 Study Guide