

# TA-002-P<sup>Q&As</sup>

HashiCorp Certified: Terraform Associate

## Pass HashiCorp TA-002-P Exam with 100% Guarantee

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

<https://www.pass2lead.com/ta-002-p.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

You have created a main.tf Terraform configuration consisting of an application server, a database, and a load balancer. You ran terraform apply and all resources were created 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 destroy the application server because it is listed first in the code
- B. Terraform will prompt you to confirm that you want to destroy all the infrastructure
- C. Terraform will destroy the main.tf file
- D. Terraform will prompt you to pick which resource you want to destroy
- E. Terraform will immediately destroy all the infrastructure

Correct Answer: B

---

### QUESTION 2

In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?

- A. Resource dependencies are identified and maintained in a file called resource.dependencies. Each terraform provider is required to maintain a list of all resource dependencies for the provider and it's included with the plugin during initialization when terraform init is executed. The file is located in the terraform.d folder.
- B. The terraform binary contains a built-in reference map of all defined Terraform resource dependencies. Updates to this dependency map are reflected in terraform versions. To ensure you are working with the latest resource dependency map you must be running the latest version of Terraform.
- C. Resource dependencies are handled automatically by the depends\_on meta\_argument, which is set to true by default.
- D. Terraform analyses any expressions within a resource block to find references to other objects, and treats those references as implicit ordering requirements when creating, updating, or destroying resources.

Correct Answer: D

<https://www.terraform.io/docs/configuration/resources.html>

---

### QUESTION 3

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. Define the attribute vnet\_id as a variable in the networking module
- B. Change the referenced value to module.my\_network.outputs.vnet\_id
- C. Define the attribute vnet\_id as an output in the networking module
- D. Change the referenced value to my\_network.outputs.vnet\_id

Correct Answer: C

In a parent module, outputs of child modules are available in expressions as module... For example, if a child module named web\_server declared an output named instance\_ip\_addr, you could access that value as module.web\_server.instance\_ip\_addr.

#### QUESTION 4

Multiple provider instances blocks for AWS can be part of a single configuration file?

- A. False
- B. True

Correct Answer: B

You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include

targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

```
# The default provider configuration
```

```
provider "aws" {  
  region = "us-east-1"  
}  
  
# Additional provider configuration for west coast region provider "aws" {  
  alias = "west"  
  region = "us-west-2"  
}
```

The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration

is considered to be the default provider configuration.

<https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances>

---

#### QUESTION 5

Terraform Enterprise (also referred to as pTFE) requires what type of backend database for a clustered deployment?

- A. PostgreSQL
- B. Cassandra
- C. MySQL
- D. MSSQL

Correct Answer: A

External Services mode stores the majority of the stateful data used by the instance in an external PostgreSQL database and an external S3-compatible endpoint or Azure blob storage. There is still critical data stored on the instance that must be managed with snapshots. Be sure to check the PostgreSQL Requirements for information that needs to be present for Terraform Enterprise to work. This option is best for users with expertise managing PostgreSQL or users that have access to managed PostgreSQL offerings like AWS RDS.

[TA-002-P PDF Dumps](#)

[TA-002-P Exam Questions](#)

[TA-002-P Braindumps](#)