

## HashiCorp

*Terraform-Associate-003*  
*HashiCorp Certified: Terraform Associate (003)*

- Up to Date products, reliable and verified.
- Questions and Answers in PDF Format.

### Full Version Features:

- 90 Days Free Updates
- 30 Days Money Back Guarantee
- Instant Download Once Purchased
- 24 Hours Live Chat Support

### For More Information:

<https://www.testsexpert.com/>

- Product Version

---

# Latest Version: 6.2

## Question: 1

Module version is required to reference a module on the Terraform Module Registry.

- A. True
- B. False

**Answer: B**

Explanation:

Module version is optional to reference a module on the Terraform Module Registry. If you omit the version constraint, Terraform will automatically use the latest available version of the module

## Question: 2

You are creating a Terraform configuration which needs to make use of multiple providers, one for AWS and one for Datadog. Which of the following provider blocks would allow you to do this?

A)

```
terraform {  
  provider "aws" {  
    profile = var.aws_profile  
    region = var.aws_region  
  }  
  
  provider "datadog" {  
    api_key = var.datadog_api_key  
    app_key = var.datadog_app_key  
  }  
}
```

B)

```
provider "aws" {  
  profile = var.aws_profile  
  region  = var.aws_region  
}  
  
provider "datadog" {  
  api_key = var.datadog_api_key  
  app_key = var.datadog_app_key  
}
```

C)

```
provider "aws" {  
  profile = var.aws_profile  
  region  = var.aws_region  
}  
  
provider "datadog" {  
  api_key = var.datadog_api_key  
  app_key = var.datadog_app_key  
}
```

D)

```
provider {  
  "aws" {  
    profile = var.aws_profile  
    region  = var.aws_region  
  }  
  
  "datadog" {  
    api_key = var.datadog_api_key  
    app_key = var.datadog_app_key  
  }  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: C**

Explanation:

Option C is the correct way to configure multiple providers in a Terraform configuration. Each provider

---

block must have a name attribute that specifies which provider it configures<sup>2</sup>. The other options are either missing the name attribute or using an invalid syntax.

### Question: 3

terraform validate confirms that your infrastructure matches the Terraform state file.

- A. True
- B. False

**Answer: B**

Explanation:

terraform validate does not confirm that your infrastructure matches the Terraform state file. It only checks whether the configuration files in a directory are syntactically valid and internally consistent<sup>3</sup>. To confirm that your infrastructure matches the Terraform state file, you need to use terraform plan or terraform apply with the -refresh-only option.

### Question: 4

Which command must you first run before performing further Terraform operations in a working directory?

- A. terraform import
- B. terraform workspace
- C. terraform plan
- D. terraform init

**Answer: D**

Explanation:

terraform init is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It initializes a working directory containing Terraform configuration files and downloads any required providers and modules. The other commands are used for different purposes, such as importing existing resources, switching between workspaces, generating execution plans, etc.

### Question: 5

A developer accidentally launched a VM (virtual machine) outside of the Terraform workflow and ended up with two servers with the same name. They don't know which VM Terraform manages but do have a list of all active VM IDs.

---

Which of the following methods could you use to discover which instance Terraform manages?

- A. Run `terraform state list` to find the names of all VMs, then run `terraform state show` for each of them to find which VM ID Terraform manages
- B. Update the code to include outputs for the ID of all VMs, then run `terraform plan` to view the outputs
- C. Run `terraform taint/code` on all the VMs to recreate them
- D. Use `terraform refresh/code` to find out which IDs are already part of state

<b>Answer: A</b>
------------------

Explanation:

The `terraform state list` command lists all resources that are managed by Terraform in the current state file<sup>1</sup>. The `terraform state show` command shows the attributes of a single resource in the state file<sup>2</sup>. By using these two commands, you can compare the VM IDs in your list with the ones in the state file and identify which one is managed by Terraform.

---

For More Information – Visit link below:  
<https://www.testsexpert.com/>

16\$ Discount Coupon: **9M2GK4NW**

## Features:

■ Money Back Guarantee.....



■ 100% Course Coverage.....



■ 90 Days Free Updates.....



■ Instant Email Delivery after Order.....

