



TA-002-P^{Q&As}

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.passapply.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

Where can Terraform not load a provider from?

- A. Plugins directory
- B. Provider plugin cache
- C. Official HashrCorp distribution on releases, hashicorp.com
- D. Source code

Correct Answer: D

QUESTION 2

While using generic git repository as a module source, which of the below options allows terraform to select a specific version or tag instead of selecting the HEAD.

- A. Append ref argument as module "vpc" { source = "git::https://example.com/vpc.git?ref=v1.2.0" }
- B. Append version argument as module "vpc" { source = "git::https://example.com/vpc.git?version=v1.2.0" }
- C. Append ref argument as module "vpc" { source = "git::https://example.com/vpc.git#ref=v1.2.0" }
- D. By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository and you can not override this.

Correct Answer: A

By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository. You can override this using the ref argument:

```
module "vpc" {  
  
source = "git::https://example.com/vpc.git?ref=v1.2.0" }
```

The value of the ref argument can be any reference that would be accepted by the git checkout command, including branch and tag names.

<https://www.terraform.io/docs/modules/sources.html>

QUESTION 3

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

- A. Destroy
- B. Apply
- C. Import



- D. Init
- E. Validate

Correct Answer: BD

Reference: <https://www.terraform.io/guides/core-workflow.html>

QUESTION 4

By default, where does Terraform store its state file?

- A. Amazon S3 bucket
- B. shared directory
- C. remotely using Terraform Cloud
- D. current working directory

Correct Answer: D

By default, the state file is stored in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

QUESTION 5

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 taint command targeting the VMs then run Terraform plan and Terraform apply
- B. Delete the Terraform VM resources from your Terraform code then run Terraform plan and terraform apply
- C. Use the terraform apply command targeting the VM resources only
- D. Use the terraform state rm command to remove the VM from state file

Correct Answer: A

<https://www.terraform.io/cli/state/taint>

QUESTION 6

What does state locking accomplish?

- A. Copies the state file from memory to disk
- B. Encrypts any credentials stored within the state file
- C. Blocks Terraform commands from modifying the state file



D. Prevents accidental deletion of the state file

Correct Answer: C

If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state. Source: <https://www.terraform.io/language/state/locking>

QUESTION 7

You need to write some Terraform code that adds 42 firewall rules to a security group as shown in the example.

```
resource "aws_security_group" "many_rules" {  
  name = "many-rules"  
  ingress {  
    from_port = 443  
    to_port = 443  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

What can you use to avoid writing 42 different nested ingress config blocks by hand?

- A. A count loop
- B. A for block
- C. A for each block
- D. A dynamic block

Correct Answer: D

A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value, and generates a nested block for each element of that complex value. Reference: <https://www.terraform.io/language/expressions/dynamic-blocks>

QUESTION 8

What is the default backend for Terraform?

- A. consul
- B. gcs



- C. local
- D. etcd

Correct Answer: C

By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to. <https://www.terraform.io/docs/backends/index.html>

QUESTION 9

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.

- A. False
- B. True

Correct Answer: B

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging. <https://www.terraform.io/docs/state/sensitive-data.html#recommendations>

QUESTION 10

Terraform plan updates your state file.

- A. True
- B. False

Correct Answer: B

The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review. Source: <https://www.terraform.io/cli/commands/plan>

QUESTION 11

Which of the below terraform commands do not run terraform refresh implicitly before taking actual action of the command?

- A. terraform apply
- B. terraform destroy
- C. terraform init



- D. terraform import
- E. terraform plan

Correct Answer: CD

<https://www.terraform.io/docs/commands/refresh.html>

QUESTION 12

How would you reference the attribute "name" of this fictitious resource in HCL?

```
resource "kubernetes_namespace" "example" {  
  name = "test"  
}
```

- A. resource.kubrnetes_namespace>example.name
- B. kubernetes_namespace.test.name
- C. kubernetes_namespace.example.name
- D. data kubernetes_namespace.name
- E. None of the above

Correct Answer: C

<https://www.terraform.io/language/expressions/references#references-to-resource-attributes>

QUESTION 13

Which of the following is true about Terraform's implementation of infrastructure as code? (Choose two.)

- A. It is only compatible with AWS infrastructure management
- B. You cannot reuse infrastructure configuration
- C. You can version your infrastructure configuration
- D. It requires manual configuration of infrastructure resources
- E. It allows you to automate infrastructure provisioning

Correct Answer: CE

QUESTION 14

A Terraform provider is not responsible for:



- A. Understanding API interactions with some service
- B. Provisioning infrastructure in multiple clouds
- C. Exposing resources and data sources based on an API
- D. Managing actions to take based on resource differences

Correct Answer: B

<https://www.terraform.io/language/providers>

QUESTION 15

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

- A. True
- B. False

Correct Answer: B

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

[TA-002-P Study Guide](#)

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