



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.geekcert.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

A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:

```
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip" }
```

What is the best way to ensure that you are not using the latest version of the module?

- A. Add a module version constraint in your configuration's backend block and specify a previous version.
- B. Add a version key to the module configuration and specify a previous version.
- C. Delete the latest version of the module in S3 to rollback to the previous version.
- D. Add a version property to the module in Terraform's state file and specify a previous version.

Correct Answer: C

Version constraints are supported only for modules installed from a module registry, such as the Terraform Registry or Terraform Cloud's private module registry. Other module sources can provide their own versioning mechanisms within the source string itself, or might not support versions at all. In particular, modules sourced from local file paths do not support version; since they're loaded from the same source repository. Only Terraform Registries support module versioning by using the version key, one cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version. <https://www.terraform.io/docs/configuration/modules.html#source>

QUESTION 2

True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

- A. False
- B. True

Correct Answer: B

The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

QUESTION 3

Only the user that generated a plan may apply it.

- A. True
- B. False



Correct Answer: B

QUESTION 4

What is a downside to using the Vault provider to read secrets from Vault?

- A. Secrets are persisted to the state file and plans.
- B. Terraform and Vault must be running on the same version.
- C. Terraform and Vault must be running on the same physical host.
- D. Terraform requires a unique auth method to work with Vault.

Correct Answer: A

The Vault provider allows Terraform to read from, write to, and configure Hashicorp Vault. Interacting with Vault from Terraform causes any secrets that you read and write to be persisted in both Terraform's state file and in any generated plan files. For any Terraform module that reads or writes Vault secrets, these files should be treated as sensitive and protected accordingly.

QUESTION 5

Terraform is currently being used by your organisation to create resources on AWS for the development of a web application. One of your coworkers wants to change the instance type to "t2.large" while keeping the default set values. What adjustments does the teammate make in order to meet his goal?

- A. Issue Terraform plan instance.type".t2.large" and it deploys the instance
- B. Modify the tf.variables with the instance type and issue terraform apply
- C. Create a new file my.tfvars and add the type of the instance and issue terraform plan and apply
- D. Modify the terraform.tfvars with the instance type and issue terraform plan and then terraform apply to deploy the instances

Correct Answer: D

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

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

[TA-002-P Practice Test](#)