

HashiCorp

TA-002-P Exam

HashiCorp Certified: Terraform Associate

**Questions & Answers
Demo**

Version: 10.0

Topic 1, Exam Set A

Question: 1

The terraform.tfstate file always matches your currently built infrastructure.

- A. True
- B. False

Answer: B

Explanation:

Reference: <https://www.terraform.io/docs/language/state/index.html>

Question: 2

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

- A. True
- B. False

Answer: B

Explanation:

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 workspaces.

Question: 3

How is the Terraform remote backend different than other state backends such as S3, Consul, etc.?

- A. It can execute Terraform runs on dedicated infrastructure on premises or in Terraform Cloud
- B. It doesn't show the output of a terraform apply locally
- C. It is only available to paying customers

D. All of the above

Answer: A

Explanation:

Backends define where Terraform's state snapshots are stored. A given Terraform configuration can either specify a backend, integrate with Terraform Cloud, or do neither and default to storing state locally.

If you and your team are using Terraform to manage meaningful infrastructure, we recommend using the remote backend with Terraform Cloud or Terraform Enterprise.

Reference: <https://www.terraform.io/docs/language/settings/backends/index.html>

Question: 4

What is the workflow for deploying new infrastructure with Terraform?

- A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
- B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.
- C. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
- D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

Answer: D

Explanation:

Reference:

<https://www.google.com/search?q=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&dq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&aqs=chrome..69i57j556j0j7&sourceid=chrome&ie=UTF-8>

Question: 5

A provider configuration block is required in every Terraform configuration.

Example:

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

- A. True
- B. False

Answer: B

Explanation:

Unlike many other objects in the Terraform language, a provider block may be omitted if its contents would otherwise be empty. Terraform assumes an empty default configuration for any provider that is not explicitly configured. <https://www.terraform.io/language/providers/configuration>