

PRACTICE QUESTIONS

HashiCorp Certified: Terraform Associate

SHARED BY [u/polite_gecko_64](#) on aiexamcerts

EDITION	Free Set 1
---------	------------

QUESTIONS	10
-----------	----

SHARED	October 15, 2026
--------	------------------

EXAM	HashiCorp Certified: Terraform Associate
------	--

Shared by [u/polite_gecko_64](#) on aiexamcerts

10 practice questions for the HashiCorp Certified: Terraform Associate exam, each followed by its answer and a short explanation. More free sets and the full practice exam:

www.aiexamcerts.com/hashicorp/hashicorp-certified-terraform-associate. Spotted a mistake? Email support@aiexamcerts.com with the exam name and the question number.

1. What does idempotence mean when an infrastructure configuration is applied repeatedly?
- A Each run deletes and recreates every resource so that the environment is rebuilt from scratch
 - B Each run is faster than the previous one because of caching
 - C Repeated runs converge on the same end state without creating duplicate resources
 - D Each run stores a new full copy of the configuration

Answer: C

An idempotent run leaves the environment unchanged once it matches the configuration, so reapplying does not multiply resources. Deleting and recreating everything on each run describes a destructive workflow, which is the tempting but wrong reading. Speed and storage of copies are unrelated to idempotence.

2. The lock file that records selected provider versions and checksums is named _____.

Answer: .terraform.lock.hcl (also accepted: terraform.lock.hcl)

The dependency lock file is written by init and should be committed so every team gets the same provider builds. The state backup file is a tempting alternative, but it stores the previous state. Both spellings with and without the leading dot are accepted.

3. Which statement best describes configuration drift?
- A Two versions of the Terraform binary report different plan output
 - B Latency grows between the CLI and the provider API
 - C A variable is declared in a module but never referenced
 - D Real infrastructure diverges from the declared configuration because of changes made outside the IaC workflow

Answer: D

Drift happens when someone changes a resource through a console or script, so reality no longer matches the code. A version mismatch between binaries is a tooling problem, which is the most tempting distractor because it also produces unexpected plans. Latency and unused variables do not change the real infrastructure.

4. Running terraform apply needs terraform init to have run first in a brand new working directory.
- A True
 - B False

Answer: A (True)

Apply needs the providers and modules that init downloads, so a fresh directory fails until init completes. Some users think apply can fetch providers on its own, which is the tempting misconception. The answer is therefore True.

5. Which two statements about remote state backends are correct? (Choose two.)
- A They remove the need for provider credentials
 - B They let a team share one state with locking to prevent concurrent applies
 - C They store the configuration files instead of the state
 - D They can encrypt state at rest in the storage service that holds it
 - E They make terraform plan skip refresh automatically

Answer: B, D

Remote backends provide shared state with locking, and many storage services encrypt data at rest. Removing provider credentials is a tempting distractor because remote runs can centralize credentials, but the backend itself does not replace them. Storing configuration and skipping refresh are not backend behaviors.

6. What does terraform plan show when it runs?
- A The actions Terraform would take to reach the desired configuration, without changing real infrastructure
 - B Only the resources currently recorded in the state file
 - C A permanent record of applied changes kept in the backend
 - D The estimated monthly cost of resources with no change details

Answer: A

Plan compares configuration, state, and refreshed real infrastructure, then shows creates, updates, and destroys without making changes. Listing only the state is the most tempting distractor, since that is what terraform state list reports. Permanent history and cost-only output are not what plan produces.

7. Terraform compares three inputs to decide what must change. Which three?
- A Variables, outputs, and locals
 - B Configuration, state, and the real infrastructure as reported by providers
 - C Modules, providers, and workspaces
 - D Plan file, lock file, and backend block

Answer: B

The configuration is the desired state, the state file records what Terraform manages, and providers report what actually exists. The variables, outputs, and locals option is tempting because all three appear in the code, but they are language constructs rather than sources of truth. The other options mix up blocks and files.

8. Inside the terraform block, the nested block that declares provider source addresses and version constraints is named _____.

Answer: required_providers

The `required_providers` block maps local provider names to their source addresses and version constraints. The `providers` block is a tempting alternative, but Terraform does not use that name for this purpose. Only the exact block name is accepted.

9. Which block reads information about existing infrastructure without creating or managing it?
- A resource
 - B module
 - C provider
 - D data

Answer: D

A data block asks a provider for information about an object that already exists, and Terraform does not manage its lifecycle. A resource block is the tempting distractor because it uses the same provider types, but a resource block creates and manages the object. Module and provider blocks configure code reuse and platform access.

10. What is the purpose of a locals block?
- A Defining named expressions that are reused within a module
 - B Declaring sensitive values supplied on the command line for every run of the configuration
 - C Registering a provider version constraint
 - D Exporting values to a parent module

Answer: A

Locals assign a name to an expression so it can be reused several times in a module without repetition. Declaring sensitive input is the tempting distractor, since that is the job of variable blocks. Version constraints live in `required_providers`, and exporting values is the job of output blocks.