



# TA-002-P<sup>Q&As</sup>

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.pass4itsure.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**

What is a key benefit of the Terraform state file?

- A. A state file represents a source of truth for resources provisioned with a public cloud console
- B. A state file represents a source of truth for resources provisioned with Terraform
- C. A state file represents the desired state expressed by the Terraform code files
- D. A state file can be used to schedule recurring infrastructure tasks

Correct Answer: C

---

**QUESTION 2**

Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?

- A. Use data source aws\_ami.
- B. Create a map of region to ami id.
- C. Create different configuration file for different region.
- D. None of the above

Correct Answer: A

<https://www.terraform.io/docs/configuration/data-sources.html>

---

**QUESTION 3**

Which are examples of infrastructure as code? (Choose two.)

- A. Cloned virtual machine images
- B. Change management database records
- C. Versioned configuration files
- D. Docker files

Correct Answer: CD

---

**QUESTION 4**

How would you be able to reference an attribute from the vsphere\_datacenter data source for use with the argument within the vsprere\_folder resource in the following configuration?



```
data "vsphere_datacenter" "dc" {}

resource "vsphere_folder" "parent" {
  path = "Production"
  type = "vm"
  datacenter_id = _____
}
```

- A. vsphere\_datacenter.dc.id
- B. data.vsphere\_datacenter.dc
- C. data.dc,id
- D. data.vsphere\_datacenter.dc.id

Correct Answer: D

#### QUESTION 5

You have declared a variable called `var.list` which is a list of objects that all have an attribute `id`. Which options will produce a list of the IDs? (Choose two.)

- A. `{ for o in var.list : o => o.id }`
- B. `var.list[*].id`
- C. `[ var.list[*].id ]`
- D. `[ for o in var.list : o.id ]`

Correct Answer: BD

<https://www.terraform.io/language/expressions/splat> A splat expression provides a more concise way to express a common operation that could otherwise be performed with a for expression.

#### QUESTION 6

What is the standard workflow that a developer follows while working with terraform open source version?

- A. Run terraform refresh to update the terraform state , then write the terraform code , and finally run terraform apply.
- B. Run terraform destroy first since you need to start from fresh every time , before running terraform apply.
- C. Write terraform code , and run terraform push , to update the terraform state to the remote repo , which in turn will take care of the next steps.
- D. Write the terraform code on the developer machine , run terraform plan to check the changes , and run terraform apply to provision the infra.



Correct Answer: D

You do not need to run terraform refresh as terraform plan implicitly will run terraform refresh.

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

---

### QUESTION 7

Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance.

Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

- A. terraform recreate
- B. terraform taint
- C. terraform destroy
- D. terraform refresh

Correct Answer: B

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply. This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change. Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run. Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case. This example will taint a single resource: `$ terraform taint aws_security_group.allow_all` The resource `aws_security_group.allow_all` in the module root has been marked as tainted. <https://www.terraform.io/docs/commands/taint.html>

---

### QUESTION 8

What is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault?

- A. Vault provider
- B. API access using the AppRole auth method
- C. integration with a tool like Jenkins
- D. CLI access from the same machine running Terraform

Correct Answer: A

---

### QUESTION 9

Which of these is the best practice to protect sensitive values in state files?



- A. Blockchain
- B. Secure Sockets Layer (SSL)
- C. Enhanced remote backends
- D. Signed Terraform providers

Correct Answer: C

Use of remote backends and especially the availability of Terraform Cloud, there are now a variety of backends that will encrypt state at rest and will not store the state in cleartext on machines running. Reference: <https://www.terraform.io/docs/extend/best-practices/sensitive-state.html>

---

#### QUESTION 10

Provisioners should only be used as a last resort.

- A. False
- B. True

Correct Answer: B

Provisioners are a Last Resort Terraform includes the concept of provisioners as a measure of pragmatism, knowing that there will always be certain behaviors that can't be directly represented in Terraform's declarative model. However, they also add a considerable amount of complexity and uncertainty to Terraform usage. Firstly, Terraform cannot model the actions of provisioners as part of a plan because they can in principle take any action. Secondly, successful use of provisioners requires coordinating many more details than Terraform usage usually requires: direct network access to your servers, issuing Terraform credentials to log in, making sure that all of the necessary external software is installed, etc. The following sections describe some situations which can be solved with provisioners in principle, but where better solutions are also available. We do not recommend using provisioners for any of the use-cases described in the following sections. Even if your specific use-case is not described in the following sections, we still recommend attempting to solve it using other techniques first, and use provisioners only if there is no other option. <https://www.terraform.io/docs/provisioners/index.html>

---

#### QUESTION 11

Select the answer below that completes the following statement: Terraform Cloud can be managed from the CLI but requires \_\_\_\_\_?

- A. an API token
- B. a TOTP token
- C. a username and password
- D. authentication using MFA

Correct Answer: A

API and CLI access are managed with API tokens, which can be generated in the Terraform Cloud UI. Each user can generate any number of personal API tokens, which allow access with their own identity and permissions. Organizations



and teams can also generate tokens for automating tasks that aren't tied to an individual user.

---

#### QUESTION 12

Which of the following can you do with terraform plan? Choose two correct answers.

- A. View the execution plan and check if the changes match your expectations
- B. Schedule Terraform to run at a planned time in the future
- C. Execute a plan in a different workspace
- D. Save a generated execution plan to apply later

Correct Answer: AD

<https://learn.hashicorp.com/tutorials/terraform/plan>

---

#### QUESTION 13

In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

- A. 5
- B. 2
- C. 3
- D. 1

Correct Answer: D

A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.  
<https://www.terraform.io/docs/cloud/workspaces/vcs.html>

---

#### QUESTION 14

Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.

- A. terraform state
- B. terraform apply
- C. terraform plan
- D. terraform validate
- E. terraform output



Correct Answer: BC

---

#### QUESTION 15

You write a new Terraform configuration and immediately run terraform apply in the CLI using the local backend. Why will the apply fail?

- A. Terraform needs you to format your code according to best practices first
- B. Terraform needs to install the necessary plugins first
- C. The Terraform CLI needs you to log into Terraform cloud first
- D. Terraform requires you to manually run terraform plan first

Correct Answer: B

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

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

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