



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

When writing Terraform code, HashiCorp recommends that you use how many spaces between each nesting level?

- A. 0
- B. 1
- C. 2
- D. 4

Correct Answer: C

The Terraform parser allows you some flexibility in how you lay out the elements in your configuration files, but the Terraform language also has some idiomatic style conventions which we recommend users always follow for consistency

between files and modules written by different teams. Automatic source code formatting tools may apply these conventions automatically.

Indent two spaces for each nesting level.

When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs:

```
ami = "abc123"
```

```
instance_type = "t2.micro"
```

When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them. Use one blank line to separate the arguments from the blocks. Use empty lines to

separate logical groups of arguments within a block. For blocks that contain both arguments and "meta-arguments" (as defined by the Terraform language semantics), list meta-arguments first and separate them from other arguments with

one blank line. Place meta-argument blocks last and separate them from other blocks with one blank line.

```
resource "aws_instance" "example" {
```

```
count = 2 # meta-argument first
```

```
ami = "abc123"
```

```
instance_type = "t2.micro"
```

```
network_interface {
```

```
# ...
```

```
}
```

```
lifecycle { # meta-argument block last
```



```
create_before_destroy = true
```

```
}  
  
}
```

Top-level blocks should always be separated from one another by one blank line. Nested blocks should also be separated by blank lines, except when grouping together related blocks of the same type (like multiple provisioner blocks in a

resource). Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example:

`root_block_device`, `ebs_block_device` and `ephemeral_block_device` on `aws_instance` form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

QUESTION 2

True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.

- A. False
- B. True

Correct Answer: A

Currently, Terraform has no mechanism to redact or protect secrets that are returned via data sources, so secrets read via this provider will be persisted into the Terraform state, into any plan files, and in some cases in the console output produced while planning and applying. These artifacts must, therefore, all be protected accordingly.

QUESTION 3

What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

- A. Local backends
- B. Providers
- C. Remote backends
- D. Workspaces

Correct Answer: D

Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
<https://www.terraform.io/docs/state/workspaces.html>



QUESTION 4

What advantage does an operations team that uses infrastructure as code have?

- A. The ability to delete infrastructure
- B. The ability to reuse best practice configurations and settings
- C. The ability to autoscale a group of servers
- D. The ability to update existing infrastructure

Correct Answer: B

QUESTION 5

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>

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

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

[TA-002-P Braindumps](#)