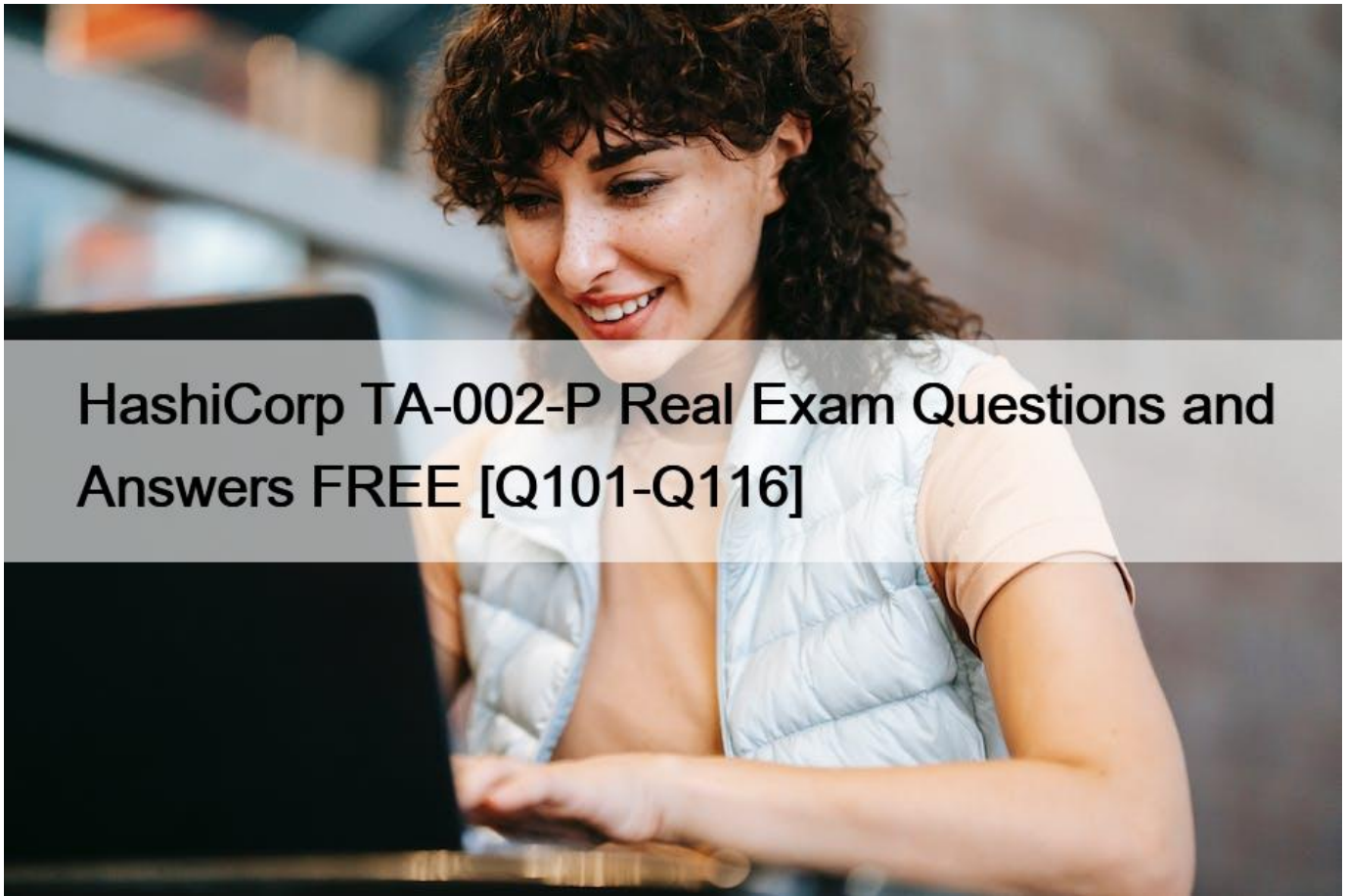


## HashiCorp TA-002-P Real Exam Questions and Answers FREE [Q101-Q116]



HashiCorp TA-002-P Real Exam Questions and Answers FREE  
Exam Dumps TA-002-P Practice Free Latest HashiCorp Practice Tests

**Q101.** Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)

- \* Unix
- \* Red Hat
- \* CentOS
- \* Amazon Linux
- \* Ubuntu

Explanation

<https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements>

**Q102.** How would you be able to reference an attribute from the vsphere\_datacenter data source for use with the argument within the vsphere\_folder resource in the following configuration?

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

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

- \* vsphere\_datacenter.dc.id
- \* data.vsphere\_datacenter.dc
- \* data.dc.id
- \* data.vsphere\_datacenter.dc.id

**Q103.** In a Terraform Cloud workspace linked to a version control repository, speculative plan runs start automatically when you merge or commit changes to version control.

- \* True
- \* False

**Q104.** When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?

- \* Object Storage
- \* State Locking
- \* WorkSpaces
- \* Encryption

Explanation

If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state.

State locking happens automatically on all operations that could write state. You won't see any message that it is happening. If state locking fails, Terraform will not continue. You can disable state locking for most commands with the -lock flag but it is not recommended.

If acquiring the lock is taking longer than expected, Terraform will output a status message. If Terraform doesn't output a message, state locking is still occurring if your backend supports it.

Not all backends support locking. Please view the list of backend types for details on whether a backend supports locking or not.

<https://www.terraform.io/docs/state/locking.html>

**Q105.** Terraform-specific settings and behaviors are declared in which configuration block type?

- \* provider
- \* terraform
- \* resource
- \* data

The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.

Example

```
terraform {  
  
  required_version = > 0.12.0;  
  
}
```

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

**Q106.** A Terraform provider is not responsible for:

- \* Understanding API interactions with some service
- \* Provisioning infrastructure in multiple clouds
- \* Exposing resources and data sources based on an API
- \* Managing actions to take based on resource differences

Explanation

<https://www.terraform.io/language/providers>

**Q107.** What is the standard workflow that a developer follows while working with terraform open source version?

- \* Run terraform refresh to update the terraform state , then write the terraform code , and finally run terraform apply.
- \* Run terraform destroy first since you need to start from fresh every time , before running terraform apply.
- \* 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.
- \* Write the terraform code on the developer machine , run terraform plan to check the changes , and run terraform apply to provision the infra.

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

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

**Q108.** You decide to move a Terraform state file to Amazon S3 from another location. You write the code below into

a file called

backend.tf ,

```
terraform {  
  backend "s3" {  
    bucket = "my-tf-bucket"  
    region = "us-east-1"  
  }  
}
```

[free.validbraindumps.com](https://www.freevalidbraindumps.com)

You immediately run terraform apply but don't see any changes. Your state file didn't move. Which command will migrate your current state file to the new S3 remote backend?

- \* terraform push
- \* terraform init
- \* terraform refresh
- \* terraform state

**Q109.** You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration?

Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.  
terraform init

Once you have authenticated to Terraform Cloud, you're ready to migrate your local state file to Terraform Cloud. To begin the migration, reinitialize. This causes Terraform to recognize your cloud block configuration.

**Q110.** You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?

- \* terraform unlock
- \* terraform force-unlock
- \* terraform state unlock
- \* None of the above

Command: force-unlock

Manually unlock the state for the defined configuration.

This will not modify your infrastructure. This command removes the lock on the state for the current configuration. The behavior of this lock is dependent on the backend being used. Local state files cannot be unlocked by another process.

<https://www.terraform.io/docs/commands/force-unlock.html>

<https://www.terraform.io/docs/state/locking.html>

Terraform has a force-unlock command to manually unlock the state if unlocking failed.

If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed.

**Q111.** Provisioners should only be used as a last resort.

- \* False
- \* True

Explanation

## 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>

### **Q112.** How can terraform plan aid in the development process?

- \* Validates your expectations against the execution plan without permanently modifying state
- \* Initializes your working directory containing your Terraform configuration files
- \* Formats your Terraform configuration files
- \* Reconciles Terraform's state against deployed resources and permanently modifies state using the current status of deployed resources

### **Q113.** terraform init initializes a sample main.tf file in the current directory.

- \* True
- \* False

### **Q114.** Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?

- \* Use terraform console command to have an interactive UI with full access to the underlying terraform state to run your interpolations , and debug at real-time.
- \* Add a breakpoint in your code, using the watch keyword , and output the value to console for temporary debugging.
- \* Use terraform zipmap function , it will be able to easily do the interpolations without complex code.
- \* Use terraform console command to have an interactive UI , but you can only use it with local state , and it does not work with remote state.

The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations

before using them in configurations, and for interacting with any values currently saved in state.

<https://www.terraform.io/docs/commands/console.html>

**Q115.** Which of the below command will upgrade the provider version to the latest acceptable one?

- \* terraform plan upgrade
- \* terraform provider -upgrade
- \* terraform init -upgrade
- \* terraform init -update

To upgrade to the latest acceptable version of each provider, run terraform init -upgrade. This command also upgrades to the latest versions of all Terraform modules.

<https://www.terraform.io/docs/configuration/providers.html>

**Q116.** Which command lets you experiment with Terraform's built-in functions?

- \* terraform env
- \* terraform console
- \* terraform test
- \* terraform validate

Explanation

<https://www.terraform.io/cli/commands/console>

HashiCorp TA-002-P (HashiCorp Certified: Terraform Associate) Certification is a valuable credential for IT professionals seeking to advance their careers in the field of infrastructure automation. HashiCorp Certified: Terraform Associate certification is designed to validate the candidate's knowledge and skills in using Terraform to automate infrastructure management tasks. Terraform is a popular open-source tool that allows users to define, provision, and manage infrastructure as code.

**Verified TA-002-P Exam Dumps Q&As - Provide TA-002-P with Correct Answers:**

<https://www.validbraindumps.com/TA-002-P-exam-prep.html>