

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.leads4pass.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

You have created a custom variable definition file `my_vars.tfvars`. How will you use it for provisioning infrastructure?

- A. `terraform apply-var-state-file="my_vars.tfvars"`
- B. `terraform apply var-file="my_vars.tfvars"`
- C. `terraform plan-var-file="my_vars.tfvar"`
- D. `terraform apply-var-file="my_vars.tfvars"`

Correct Answer: D

To set lots of variables, it is more convenient to specify their values in a variable definitions file (with a filename ending in either `.tfvars` or `.tfvars.json`) and then specify that file on the command line with `-var-file`:

```
terraform apply-var-file="my_vars.tfvars"
```

<https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files>

QUESTION 2

Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)

- A. GitHub
- B. CVS Version Control
- C. Azure DevOps Server
- D. Bitbucket Cloud
- E. GitHub Enterprise

Correct Answer: ACDE

Terraform Cloud supports the following VCS providers: -<https://www.terraform.io/docs/cloud/vcs/github.html>
-<https://www.terraform.io/docs/cloud/vcs/github.html> -<https://www.terraform.io/docs/cloud/vcs/github-enterprise.html>
-<https://www.terraform.io/docs/cloud/vcs/gitlab-com.html> -<https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html>
-<https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html> -<https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html> -<https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html>
-<https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html>
<https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers>

QUESTION 3

You have multiple developers working on a terraform project (using terraform OSS), and have saved the terraform state in a remote S3 bucket . However ,team is intermittently experiencing inconsistencies in the provisioned infrastructure / failure in the code . You have traced this problem to simultaneous/concurrent runs of terraform apply command for

2/more developers . What can you do to fix this problem?

- A. Use terraform workspaces feature, this will fix this problem by default , as every developer will have their own state file , and terraform will merge them on server side on its own.
- B. Structure your team in such a way that only one individual will run terraform apply , everyone will just make changes and share with him. Then there will be no chance of any inconsistencies.
- C. Stop using remote state , and store the developer tfstate in their own machine . Once a day , all developers should sit together and merge the state files manually , to avoid any inconsistencies.
- D. Enable terraform state locking for the S3 backend using DynamoDB table. This prevents others from acquiring the lock and potentially corrupting your state.

Correct Answer: D

S3 backend support state locking using DynamoDB. <https://www.terraform.io/docs/state/locking.html>

QUESTION 4

Multiple configurations for the same provider can be used in a single configuration file.

- A. False
- B. True

Correct Answer: B

You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include

targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

```
# The default provider configuration
```

```
provider "aws" {  
  region = "us-east-1"  
}
```

```
# Additional provider configuration for west coast region provider "aws" {
```

```
  alias = "west"  
  region = "us-west-2"  
}
```

The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty

configuration

is considered to be the default provider configuration.

<https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances>

QUESTION 5

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.

A. False

B. True

Correct Answer: B

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging. <https://www.terraform.io/docs/state/sensitive-data.html#recommendations>

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

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

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