

TA-003-P^{Q&As}

HashiCorp Certified: Terraform Associate

Pass HashiCorp TA-003-P Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.leads4pass.com/ta-003-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

Which Terraform collection type should you use to store key/value pairs?

- A. Set
- B. Map
- C. Tuple
- D. list

Correct Answer: B

The Terraform collection type that should be used to store key/value pairs is map. A map is a collection of values that are accessed by arbitrary labels, called keys. The keys and values can be of any type, but the keys must be unique within a map. For example, `var = { key1 = "value1", key2 = "value2" }` is a map with two key/value pairs. Maps are useful for grouping related values together, such as configuration options or metadata. References = [Collection Types], [Map Type Constraints]

QUESTION 2

As a developer, you want to ensure your plugins are up to date with the latest versions. Which Terraform command should you use?

- A. `terraform refresh -upgrade`
- B. `terraform apply -upgrade`
- C. `terraform init -upgrade`
- D. `terraform providers -upgrade`

Correct Answer: C

This command will upgrade the plugins to the latest acceptable version within the version constraints specified in the configuration. The other commands do not have an `- upgrade` option.

QUESTION 3

A developer on your team is going to tear down an existing deployment managed by Terraform and deploy a new one. However, there is a server resource named `aws_instance.ubuntu[1]` they would like to keep. What command should they use to tell Terraform to stop managing that specific resource?

- A. `Terraform plan rm:aws_instance.ubuntu[1]`
- B. `Terraform state rm:aws_instance.ubuntu[1]`
- C. `Terraform apply rm:aws_instance.ubuntu[1]`
- D. `Terraform destroy rm:aws_instance.ubuntu[1]`

Correct Answer: B

To tell Terraform to stop managing a specific resource without destroying it, you can use the terraform state rm command. This command will remove the resource from the Terraform state, which means that Terraform will no longer track or update the corresponding remote object. However, the object will still exist in the remote system and you can later use terraform import to start managing it again in a different configuration or workspace. The syntax for this command is terraform state rm , where is the resource address that identifies the resource instance to remove. For example, terraform state rm aws_instance.ubuntu[1] will remove the second instance of the aws_instance resource named ubuntu from the state. References = : Command: state rm : Moving Resources

QUESTION 4

What information does the public Terraform Module Registry automatically expose about published modules?

- A. Required input variables
- B. Optional inputs variables and default values
- C. Outputs
- D. All of the above
- E. None of the above

Correct Answer: D

The public Terraform Module Registry automatically exposes all the information about published modules, including required input variables, optional input variables and default values, and outputs. This helps users to understand how to use and configure the modules.

QUESTION 5

FILL IN THE BLANK

What is the name of the default file where Terraform stores the state?

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

Correct Answer: Terraform.tfstate

Terraform.tfstate

The name of the default file where Terraform stores the state is terraform.tfstate. This file contains a JSON representation of the current state of the infrastructure managed by Terraform. Terraform uses this file to track the metadata and

attributes of the resources, and to plan and apply changes. By default, Terraform stores the state file locally in the same directory as the configuration files, but it can also be configured to store the state remotely in a backend. References =

[Terraform State], [State File Format]

QUESTION 6

When should you write Terraform configuration files for existing infrastructure that you want to start managing with Terraform?

- A. You can import infrastructure without corresponding Terraform code
- B. Terraform will generate the corresponding configuration files for you
- C. Before you run terraform Import
- D. After you run terraform import

Correct Answer: C

You need to write Terraform configuration files for the existing infrastructure that you want to import into Terraform, otherwise Terraform will not know how to manage it. The configuration files should match the type and name of the resources that you want to import.

QUESTION 7

When should you run terraform init?

- A. Every time you run terraform apply
- B. Before you start coding a new Terraform project
- C. After you run terraform plan for the time in a new terraform project and before you run terraform apply
- D. After you start coding a new terraform project and before you run terraform plan for the first time.

Correct Answer: D

You should run terraform init after you start coding a new Terraform project and before you run terraform plan for the first time. This command will initialize the working directory by downloading the required providers and modules, creating the initial state file, and performing other necessary tasks. References = : Initialize a Terraform Project

QUESTION 8

What are some benefits of using Sentinel with Terraform Cloud/Terra form Cloud? Choose three correct answers.

- A. You can enforce a list of approved AWS AMIs
- B. Policy-as-code can enforce security best practices
- C. You can check out and check in cloud access keys
- D. You can restrict specific resource configurations, such as disallowing the use of CIDR=0.0.0.0/0.
- E. Sentinel Policies can be written in HashiCorp Configuration Language (HCL)

Correct Answer: ABD

These are some of the benefits of using Sentinel with Terraform Cloud/Terraform Enterprise, as they allow you to implement logic-based policies that can access and evaluate the Terraform plan, state, and configuration. The other options are not true, as Sentinel does not manage cloud access keys, and Sentinel policies are written in Sentinel language, not HCL.

QUESTION 9

You are working on some new application features and you want to spin up a copy of your production deployment to perform some quick tests. In order to avoid having to configure a new state backend, what open source Terraform feature would allow you create multiple states but still be associated with your current code?

- A. Terraform data sources
- B. Terraform local values
- C. Terraform modules
- D. Terraform workspaces
- E. None of the above

Correct Answer: D

Terraform workspaces allow you to create multiple states but still be associated with your current code. Workspaces are like "environments" (e.g. staging, production) for the same configuration. You can use workspaces to spin up a copy of your production deployment for testing purposes without having to configure a new state backend. Terraform data sources, local values, and modules are not features that allow you to create multiple states. References = Workspaces and How to Use Terraform Workspaces

QUESTION 10

You can configure Terraform to log to a file using the TF_LOG environment variable.

- A. True
- B. False

Correct Answer: A

You can configure Terraform to log to a file using the TF_LOG environment variable. This variable can be set to one of the log levels: TRACE, DEBUG, INFO, WARN or ERROR. You can also use the TF_LOG_PATH environment variable to specify a custom log file location. References = : Debugging Terraform

QUESTION 11

Your security team scanned some Terraform workspaces and found secrets stored in plaintext in state files. How can you protect that data?

- A. Edit your state file to scrub out the sensitive data
- B. Always store your secrets in a secrets.tfvars file

- C. Delete the state file every time you run Terraform
- D. Store the state in an encrypted backend

Correct Answer: D

This is a secure way to protect sensitive data in the state file, as it will be encrypted at rest and in transit². The other options are not recommended, as they could lead to data loss, errors, or security breaches.

QUESTION 12

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 correct answers.

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

Correct Answer: BC

These are two ways to produce a list of the IDs from a list of objects that have an attribute `id`, using either a `for` expression or a `splat` expression syntax.

QUESTION 13

If a DevOps team adopts AWS CloudFormation as their standardized method for provisioning public cloud resources, which of the following scenarios poses a challenge for this team?

- A. The team is asked to manage a new application stack built on AWS-native services
- B. The organization decides to expand into Azure wishes to deploy new infrastructure
- C. The team is asked to build a reusable code based that can deploy resources into any AWS region
- D. The DevOps team is tasked with automating a manual, web console-based provisioning.

Correct Answer: B

This is the scenario that poses a challenge for this team, if they adopt AWS CloudFormation as their standardized method for provisioning public cloud resources, as CloudFormation only supports AWS services and resources, and cannot be used to provision infrastructure on other cloud platforms such as Azure.

QUESTION 14

What does Terraform not reference when running a `terraform apply -refresh-only` ?

- A. State file

- B. Credentials
- C. Cloud provider
- D. Terraform resource definitions in configuration files

Correct Answer: D

When running a terraform apply -refresh-only, Terraform does not reference the configuration files, but only the state file, credentials, and cloud provider. The purpose of this command is to update the state file with the current status of the real resources, without making any changes to them1.

QUESTION 15

You add a new resource to an existing Terraform configuration, but do not update the version constraint in the configuration. The existing and new resources use the same provider. The working contains a .terraform.lock, hc1 file. How will Terraform choose which version of the provider to use?

- A. Terraform will use the version recorded in your lock file
- B. Terraform will use the latest version of the provider for the new resource and the version recorded in the lock file to manage existing resources
- C. Terraform will check your state file to determine the provider version to use
- D. Terraform will use the latest version of the provider available at the time you provision your new resource

Correct Answer: A

This is how Terraform chooses which version of the provider to use, when you add a new resource to an existing Terraform configuration, but do not update the version constraint in the configuration. The lock file records the exact version of each provider that was installed in your working directory, and ensures that Terraform will always use the same provider versions until you run terraform init -upgrade to update them.

[Latest TA-003-P Dumps](#)

[TA-003-P VCE Dumps](#)

[TA-003-P Practice Test](#)