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.pass2lead.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

Refer to the following terraform variable definition

variable "track_tag" { type = list default = ["data_ec2","integration_ec2","digital_ec2"]} track_tag = { Name = element(var.track_tag,count.index)}

If count.index is set to 2, which of the following values will be assigned to the name attribute of track_tag variable?

A. integration_ec2

B. digital_ec2

C. track_tag

D. data_ec2

Correct Answer: B

QUESTION 2

Mary has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code:

1.

output "db_password"

2.

{

3.

value = local.db_password

4.

}

Mary wants to hide the output value in the CLI after terraform apply? What is the best way?

A. Use secure parameter

B. Use sensitive parameter

C. Use cryptographic hash

D. Encrypt the value using encrypt() function

Correct Answer: B

QUESTION 3

A Terraform provisioner must be nested inside a resource configuration block.

A. True

B. False

Correct Answer: A

Most provisioners require access to the remote resource via SSH or WinRM, and expect a nested connection block with details about how to connect. Reference:
https://www.terraform.io/docs/language/resources/provisioners/connection.html

QUESTION 4

You have created a main.tr Terraform configuration consisting of an application server, a database, and a load balancer. You ran terraform apply and all resources were created successfully. Now you realize that you do not actually need the load balancer so you run terraform destroy without any flags

What will happen?

A. Terraform will destroy the application server because it is listed first in the code

B. Terraform will prompt you to confirm that you want to destroy all the infrastructure

C. Terraform will destroy the main.tf file

D. Terraform will prompt you to pick which resource you want to destroy

E. Terraform will immediately destroy all the infrastructure

Correct Answer: B

QUESTION 5

Any user can publish modules to the public Terraform Module Registry.

A. True

B. False

Correct Answer: B

QUESTION 6

True or False: A list(...) contain a number of values of the same type while an object(...) can contain a number of values of different types.

A. False

B. True

Correct Answer: B

Collection Types

A collection type allows multiple values of one other type to be grouped together as a single value. The type of value within a collection is called its element type. All collection types must have an element type, which is provided as the

argument to their constructor. For example, the type list(string) means "list of strings", which is a different type than list(number), a list of numbers. All elements of a collection must always be of the same type.

The three kinds of collection type in the Terraform language are:

*

list(...): a sequence of values identified by consecutive whole numbers starting with zero. The keyword list is a shorthand for list(any), which accepts any element type as long as every element is the same type. This is for compatibility with

older configurations; for new code, we recommend using the full form.

*

map(...): a collection of values where each is identified by a string label. The keyword map is a shorthand for map(any), which accepts any element type as long as every element is the same type. This is for compatibility with older

configurations; for new code, we recommend using the full form.

*

set(...): a collection of unique values that do not have any secondary identifiers or ordering.

https://www.terraform.io/docs/configuration/types.html Structural Types

A structural type allows multiple values of several distinct types to be grouped together as a single value. Structural types require a schema as an argument, to specify which types are allowed for which elements.

The two kinds of structural type in the Terraform language are:

* object(...): a collection of named attributes that each have their own type. The schema for object types is { = , = , ... }--a pair of curly braces containing a comma-separated series of = pairs.

Values that match the object type must contain all of the specified keys, and the value for each key must match its specified type. (Values with additional keys can still match an object type, but the extra attributes are discarded during type

conversion.)

*

tuple(...): a sequence of elements identified by consecutive whole numbers starting with zero, where each element has its own type. The schema for tuple types is [, , ...]--a pair of square brackets containing a comma-

separated series of types. Values that match the tuple type must have exactly the same number of elements (no more and no fewer), and the value in each position must match the specified type for that position.

For example: an object type of object({ name=string, age=number }) would match a value like the following:

{

name = "John"

age = 52

}

Also, an object type of object({ id=string, cidr_block=string }) would match the object produced by a reference to an aws_vpc resource, like aws_vpc.example_vpc; although the resource has additional attributes, they would be discarded

during type conversion. Finally, a tuple type of tuple([string, number, bool]) would match a value like the following:

["a", 15, true]

https://www.terraform.io/docs/configuration/types.html

QUESTION 7

What Terraform feature is shown in the example below?

A. conditional expression

B. local values

C. dynamic block

D. data source

Correct Answer: C

QUESTION 8

What kind of resource dependency is stored in terraform.tfstate file?

A. Both implicit and explicit dependencies are stored in state file.

B. Only explicit dependencies are stored in state file.

C. Only implicit dependencies are stored in state file.

D. No dependency information is stored in state file.

Correct Answer: A

Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state.

https://www.terraform.io/docs/state/purpose.html#metadata

QUESTION 9

When Terraform needs to be installed in a location where it does not have internet access to download the installer and upgrades, the installation is generally known as to be _____.

A. a private install

B. disconnected

C. air-gapped

D. non-traditional

Correct Answer: D

A Terraform Enterprise install that is provisioned on a network that does not have Internet access is generally known as an air-gapped install. These types of installs require you to pull updates, providers, etc. from external sources vs. being able to download them directly.

QUESTION 10

After creating a new workspace "PROD" you need to run the command terraform select PROD to switch to it.

A. False

B. True

Correct Answer: A

By default, when you create a new workspace you are automatically switched to it To create a new workspace and switch to it, you can use terraform workspace new ; to switch to a existing workspace you can use

terraform workspace select ;

Example:

$ terraform workspace new example

Created and switched to workspace "example"!

You\\'re now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.

QUESTION 11

Which of the following command can be used to view the specified version constraints for all providers used in the current configuration.

A. terraform providers

B. terraform state show

C. terraform provider

D. terraform plan

Correct Answer: A

Use the terraform providers command to view the specified version constraints for all providers used in the current configuration. https://www.terraform.io/docs/configuration/providers.html

QUESTION 12

You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season. Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.

How would you solve this using infrastructure as code?

A. Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource

B. Implement a checklist that engineers can follow when configuring compute instances

C. Replace the compute instance type with a larger version to reduce the number of required deployments

D. Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews

Correct Answer: D

QUESTION 13

By default, provisioners that fail will also cause the Terraform apply itself to error. How can you change this default behavior within a provisioner?

A. provisioner "local-exec" { on_failure = "next" }

B. provisioner "local-exec" { when = "failure" terraform apply }

C. provisioner "local-exec" { on_failure = "continue" }

D. provisioner "local-exec" { on_failure = continue }

Correct Answer: C

https://www.terraform.io/docs/provisioners/index.html

QUESTION 14

What is the purpose of using the local-exec provisioner? (Select Two)

A. To invoke a local executable.

B. Executes a command on the resource to invoke an update to the Terraform state.

C. To execute one or more commands on the machine running Terraform.

D. Ensures that the resource is only executed in the local infrastructure where Terraform is deployed.

Correct Answer: AC

The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource. Note that even though the resource will be fully created when the provisioner is

run, there is no guarantee that it will be in an operable state-for example system services such as sshd may not be started yet on compute resources.

Example usage

resource "aws_instance" "web" {

...

provisioner "local-exec" {

command = "echo ${aws_instance.web.private_ip} >> private_ips.txt" }

}

Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives.

https://www.terraform.io/docs/provisioners/local-exec.html

QUESTION 15

Which one is the right way to import a local module names consul?

A. module "consul" { source = "consul"}

B. module "consul" { source = "./consul"}

C. module "consul" { source = "../consul"}

D. module "consul" { source = "module/consul"}

Correct Answer: BC

A local path must begin with either ./ or ../ to indicate that a local path is intended, to distinguish from a module registry address.

module "consul" {

source = "./consul"

}

TA-002-P VCE Dumps TA-002-P Exam Questions TA-002-P Braindumps