Free Questions for VA-002-P by ebraindumps

Shared by Ortiz on 06-06-2022

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which statements best describes what the local variable assignment is doing in the following code snippet:

```
variable "subnet_details" {
type = list(object({
cidr = string
subnet_name = string
route_table_name = string
aznum = number
}))
}
locals {
route_tables_all = distinct([for s in var.subnet_details : s.route_table_name ])
}
```

Options:

A- Create a distinct list of route table name objects

B- Create a map of route table names to subnet names

C- Create a map of route table names from a list of subnet names

D- Create a list of route table names eliminating duplicates

Answer:

D

Explanation:

route_tables_all is assigned a list of unique route table names filtered from a list of objects describing subnet details, one of those object attributes being route_table_name.

Question 2

Question Type: MultipleChoice

From the code below, identify the implicit dependency:

```
resource "aws_eip" "public_ip" {
vpc = true
instance = aws_instance.web_server.id
}
resource "aws_instance" "web_server" {
ami = "ami-2757f631"
instance_type = "t2.micro"
depends_on = [aws_s3_bucket.company_data]
}
```

Options:

A- The EC2 instance labeled web_server

B- The EIP with an id of ami-2757f631

C- The AMI used for the EC2 instance

D- The S3 bucket labeled company_data

Answer:

A

Explanation:

The EC2 instance labeled web_server is the implicit dependency as the aws_eip cannot be created until the aws_instance labeled web_server has been provisioned and the id is available.

Note that aws_s3_bucket.example is an explicit dependency.

Question 3

Question Type: **MultipleChoice**

Environment variables can be used to set variables. The environment variables must be in the format "____"_. Select the correct prefix string from the following list.

Options:

A- TF_VAR

B- TF_VAR_NAME

C- TF_ENV

D- TF_ENV_VAR

Answer:

A

Explanation:

Environment variables can be used to set variables. The environment variables must be in the format TF_VAR_name and this will be checked last for a value. For example:

```
export TF_VAR_region=us-west-1

export TF_VAR_ami=ami-049d8641

export TF_VAR_alist='[1,2,3]'

export TF_VAR_amap='{ foo = 'bar', baz = 'qux' }'
```

https://www.terraform.io/docs/commands/environment-variables.html

Question 4

Question Type: MultipleChoice

Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements. (select three)

Options:

A- The registry uses tags to identify module versions. Release tag names must be for the format x.y.z, and can optionally be prefixed with a v.

B- Module repositories must use this three-part name format, terraform-<PROVIDER>-<NAME>.

C- The module must be PCI/HIPPA compliant.

D- The module must be on GitHub and must be a public repo

Answer:

A, B, D

Explanation:

The list below contains all the requirements for publishing a module. Meeting the requirements for publishing a module is extremely easy. The list may appear long only to ensure we're detailed, but adhering to the requirements should happen naturally.

GitHub. The module must be on GitHub and must be a public repo. This is only a requirement for the public registry. If you're using a private registry, you may ignore this requirement.

Named terraform-<PROVIDER>-<NAME>. Module repositories must use this three-part name format, where <NAME> reflects the type of infrastructure the module manages, and <PROVIDER> is the main provider where it creates that infrastructure. The <NAME> segment can contain additional hyphens. Examples: terraform-google-vault or terraform-aws-ec2-instance.

Repository description. The GitHub repository description is used to populate the short description of the module. This should be a simple one-sentence description of the module.

Standard module structure. The module must adhere to the standard module structure. This allows the registry to inspect your module and generate documentation, track resource usage, parse submodules and examples, and more.

x.y.z tags for releases. The registry uses tags to identify module versions. Release tag names must be a semantic version, which can optionally be prefixed with a v. For example, v1.0.4 and 0.9.2. To publish a module initially, at least one release tag must be present. Tags that don't look like version numbers are ignored.

https://www.terraform.io/docs/registry/modules/publish.html#requirements

Question 5

Question Type: MultipleChoice

What is the result of the following terraform function call?

lookup({a="hello", b="goodbye"}, "c", "what?")

Options:

A- goodbye

B- hello

C- what?

D- c

Answer:

C

Explanation:

lookup retrieves the value of a single element from a map, given its key. If the given key does not exist, the given default value is returned instead. In this case, the function call is searching for the key 'c'. But since there is no key 'c', the default vault 'what?' is returned.

https://www.terraform.io/docs/configuration/functions/lookup.html

Question 6

Question Type: **MultipleChoice**

Which of the following statements best describes the Terraform list(...) type?

Options:

A- a collection of unique values that do not have any secondary identifiers or ordering.

B- a collection of values where each is identified by a string label.

C- a sequence of values identified by consecutive whole numbers starting with zero.

D- a collection of named attributes that each have their own type.

Answer:

C

Explanation:

A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.

https://www.terraform.io/docs/configuration/types.html#structural-types

Question 7

Question Type: MultipleChoice

True or False?

terraform init cannot automatically download Community providers.

Options:

A- False

B- True

Answer:

B

Explanation:

Anyone can develop and distribute their own Terraform providers. (See Writing Custom Providers for more about provider development.) These third-party providers must be manually installed, since terraform init cannot automatically download them.

https://www.terraform.io/docs/configuration/providers.html#third-party-plugins