Free Questions for Terraform-Associate-003 by braindumpscollection

Shared by Marshall on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

What value does the Terraform Cloud private registry provide over the public Terraform Module Registry?

Options:

A- The ability to share modules publicly with any user of Terraform

B- The ability to restrict modules to members of Terraform Cloud or Enterprise organizations

C- The ability to tag modules by version or release

D- The ability to share modules with public Terraform users and members of Terraform Cloud Organizations

Answer:

B

Explanation:

The Terraform Cloud private registry provides the ability to restrict modules to members of Terraform Cloud or Enterprise organizations. This allows you to share modules within your organization without exposing them to the public. The private registry also supports importing modules from your private VCS repositories. The public Terraform Module Registry, on the other hand, publishes modules

from public Git repositories and makes them available to any user of Terraform.Reference= :Private Registry - Terraform Cloud:Terraform Registry - Provider Documentation

Question 2

Question Type: MultipleChoice

When should you run terraform init?

Options:

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.

Answer:

D

Explanation:

You should runterraform initafter you start coding a new Terraform project and before you runterraform planfor 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.Reference= : Initialize a Terraform Project

Question 3

Question Type: MultipleChoice

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

Options:

A- True

B- False

Answer:

A

Explanation:

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.Reference= : Debugging Terraform

Question 4

Question Type: MultipleChoice

You have used Terraform lo create an ephemeral development environment in the (loud and are now ready to destroy all the Infrastructure described by your Terraform configuration To be safe, you would like to first see all the infrastructure that Terraform will delete.

Which command should you use to show all of the resources that mil be deleted? Choose two correct answers.

Options:

A- Run terraform state rm '

B- Run terraform show :destroy

C- Run terraform destroy and it will first output all the resource that will be deleted before prompting for approval

D- Run terraform plan .destory

Answer:

C, D

Explanation:

To see all the resources that Terraform will delete, you can use either of these two commands:

terraform destroywill show the plan of destruction and ask for your confirmation before proceeding. You can cancel the command if you do not want to destroy the resources.

terraform plan -destroywill show the plan of destruction without asking for confirmation. You can use this command to review the changes before runningterraform destroy.Reference= : Destroy Infrastructure : Plan Command: Options

Question 5

Question Type: MultipleChoice

Which configuration consistency errors does terraform validate report?

Options:

A- Terraform module isn't the latest version

B- Differences between local and remote state

C- Declaring a resource identifier more than once

D- A mix of spaces and tabs in configuration files

Answer:

C

Explanation:

Terraform validate reports configuration consistency errors, such as declaring a resource identifier more than once. This means that the same resource type and name combination is used for multiple resource blocks, which is not allowed in Terraform. For example,resource 'aws_instance' 'example' {...}cannot be used more than once in the same configuration. Terraform validate does not report errors related to module versions, state differences, or formatting issues, as these are not relevant for checking the configuration syntax and structure.Reference= [Validate Configuration], [Resource Syntax]

Question 6

Question Type: MultipleChoice

You are writing a child Terraform module that provisions an AWS instance. You want to reference the IP address returned by the child module in the root configuration. You name the instance resource "main'.

Which of these is the correct way to define the output value?

A)

```
output "instance_ip_addr" {
  return aws_instance.main.private_ip
}
```

B)

```
output "aws_instance.instance_ip_addr" {
  return aws_instance.main.private_ip
}
```

C)

```
output "aws_instance.instance_ip_addr" {
  value = ${main.private_ip}
}
```

D)

```
output "instance_ip_addr" {
  value = aws_instance.main.private_ip
}
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

D

Question 7

Question Type: **MultipleChoice**

Terraform configuration can only import modules from the public registry.

Options:

A- True

B- False

Answer:

B

Explanation:

Terraform configuration can import modules from various sources, not only from the public registry. Modules can be sourced from local file paths, Git repositories, HTTP URLs, Mercurial repositories, S3 buckets, and GCS buckets. Terraform supports a number of common conventions and syntaxes for specifying module sources, as documented in the [Module Sources] page.Reference= [Module Sources]

Question 8

Question Type: **MultipleChoice**

Which is the best way to specify a tag of v1.0.0 when referencing a module stored in Git (for example.

Git::https://example.com/vpc.git)?

Options:

A- Append pref=v1.0.0 argument to the source path

B- Add version = "1.0.0" parameter to module block

C- Nothing modules stored on GitHub always default to version 1.0.0

Answer:

A

Explanation:

The best way to specify a tag of v1.0.0 when referencing a module stored in Git is to append?ref=v1.0.0argument to the source path. This tells Terraform to use a specific Git reference, such as a branch, tag, or commit, when fetching the module source code. For example,source = 'git::https://example.com/vpc.git?ref=v1.0.0'. This ensures that the module version is consistent and reproducible across different environments.Reference= [Module Sources], [Module Versions]

Question 9

Question Type: MultipleChoice

When should you use the force-unlock command?

Options:

A- You have a high priority change

B- Automatic unlocking failed

C- apply failed due to a state lock

D- You see a status message that you cannot acquire the lock

Answer:

B

Explanation:

You should use the force-unlock command when automatic unlocking failed. Terraform will lock your state for all operations that could write state, such as plan, apply, or destroy. This prevents others from acquiring the lock and potentially corrupting your state. State locking happens automatically on all operations that could write state and 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. Terraform has a force-unlock command to manually unlock the state if unlocking failed. Be very careful with this command. 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. To protect you, the force-unlock command requires a unique lock ID. Terraform will output this lock ID if unlocking fails. This lock ID acts as a nonce, ensuring that locks and unlocks target the correct lock. The other situations are not valid reasons to use the force-unlock command. You should not use the force-unlock command if you have a high priority change, if apply failed due to a state lock, or if you see a status message that you cannot acquire the lock. These situations indicate that someone else is holding the lock and you should wait for them to finish their operation or contact them to resolve the issue. Using the force-unlock command in these cases could result in data loss or inconsistency.Reference= [State Locking], [Command: force-unlock]