HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

About Exambible

Your Partner of IT Exam

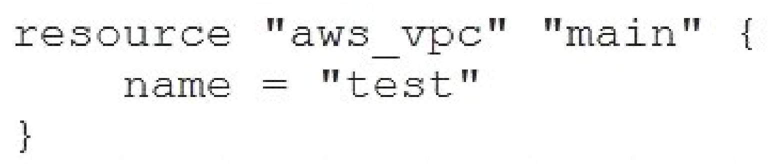
Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime
  All examinations will be up to date.
* 24/7 Quality Support
  We will provide service round the clock.
* 100% Pass Rate
  Our guarantee that you will pass the exam.
* Unique Gurantee
  If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1
- (Exam Topic 1)
What is the provider for this fictitious resource?



```
resource "aws_vpc" "main" {
    name = "test"
}
```

A. vpc
B. main
C. aws
D. test

Answer: C

Explanation:

Reference: https://docs.aws.amazon.com/cloudformation-cli/latest/userguide/resource-types.html

NEW QUESTION 2
- (Exam Topic 1)
Which option can not be used to keep secrets out of Terraform configuration files?

A. A Terraform provider
B. Environment variables
C. A -var flag
D. secure string

Answer: A

Explanation:

Reference: https://secrethub.io/blog/secret-management-for-terraform/

NEW QUESTION 3
- (Exam Topic 1)
When should you use the force-unlock command?

A. You see a status message that you cannot acquire the lock
B. You have a high priority change
C. Automatic unlocking failed
D. Your apply failed due to a state lock

Answer: C

Explanation:
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. Source: https://www.terraform.io/language/state/locking
https://www.terraform.io/cli/commands/force-unlock

NEW QUESTION 4
- (Exam Topic 1)
Terraform provisioners can be added to any resource block.

A. True
B. False

Answer: A

Explanation:
https://www.phillipsj.net/posts/introduction-to-terraform-provisioners/
As you continue learning about Terraform, you will start hearing about provisioners. Terraform provisioners can be created on any resource and provide a way to execute actions on local or remote machines.
https://www.terraform.io/language/resources/provisioners/local-exec

NEW QUESTION 5
- (Exam Topic 1)
Which argument(s) is (are) required when declaring a Terraform variable?

A. type
B. default
C. description
D. All of the above
E. None of the above

Answer: B

Explanation:
The variable declaration can also include a default argument.
Reference: https://www.terraform.io/docs/language/values/variables.html

NEW QUESTION 6
- (Exam Topic 1)
Why would you use the terraform taint command?

A. When you want to force Terraform to destroy a resource on the next apply
B. When you want to force Terraform to destroy and recreate a resource on the next apply
C. When you want Terraform to ignore a resource on the next apply
D. When you want Terraform to destroy all the infrastructure in your workspace

Answer: B

Explanation:
The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
Reference: https://www.terraform.io/docs/cli/commands/taint.html

NEW QUESTION 7
- (Exam Topic 1)
You're building a CI/CD (continuous integration/ continuous delivery) pipeline and need to inject sensitive variables into your Terraform run.
How can you do this safely?

A. Pass variables to Terraform with a –var flag
B. Copy the sensitive variables into your Terraform code
C. Store the sensitive variables in a secure_vars.tf file
D. Store the sensitive variables as plain text in a source code repository

Answer: A

Explanation:
https://blog.gruntwork.io/a-comprehensive-guide-to-managing-secrets-in-your-terraform-code-1d586955ace1

NEW QUESTION 8
- (Exam Topic 1)
Terraform provisioners that require authentication can use the _____ block.

A. connection
B. credentials
C. secrets
D. ssh

Answer: A

Explanation:
https://www.terraform.io/language/resources/provisioners/connection
"Most provisioners require access to the remote resource via SSH or WinRM and expect a nested connection block with details about how to connect."
"Connection blocks don't take a block label and can be nested within either a resource or a provisioner."

NEW QUESTION 9
- (Exam Topic 1)
Which of the following is allowed as a Terraform variable name?

A. count
B. name
C. source
D. version

Answer: B

Explanation:
"The name of a variable can be any valid identifier except the following: source, version, providers, count, for_each, lifecycle, depends_on, locals."
https://www.terraform.io/language/values/variables

NEW QUESTION 10
- (Exam Topic 1)
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.)

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

Answer: BD

Explanation:
https://www.terraform.io/language/expressions/splat
A splat expression provides a more concise way to express a common operation that could otherwise be performed with a for expression.

NEW QUESTION 10
- (Exam Topic 1)
What is the workflow for deploying new infrastructure with Terraform?

A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.
C. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

Answer: D

Explanation:
Reference: https://www.google.com/search?q=Write+a+Terraform+configuration%2C+run+terraform+init%2C
+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new
+infrastructure.&oq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to
+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new
+infrastructure.&aqs=chrome..69i57.556j0j7&sourceid=chrome&ie=UTF-8

NEW QUESTION 12
- (Exam Topic 1)
Terraform variables and outputs that set the "description" argument will store that description in the state file.

A. True
B. False

Answer: B

Explanation:
Reference: https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION 14
- (Exam Topic 1)
What is not processed when running a terraform refresh?

A. State file
B. Configuration file
C. Credentials
D. Cloud provider

Answer: B

Explanation:
"The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match."

NEW QUESTION 16
- (Exam Topic 1)
A Terraform provisioner must be nested inside a resource configuration block.

A. True
B. False

Answer: A

Explanation:
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

NEW QUESTION 21
- (Exam Topic 1)
In Terraform 0.13 and above, outside of the required_providers block, Terraform configurations always refer to providers by their local names.

A. True
B. False

Answer: A

Explanation:
Outside of the required_providers block, Terraform configurations always refer to providers by their local names.
Reference: https://www.terraform.io/docs/language/providers/requirements.html https://www.terraform.io/language/providers/requirements#local-names

NEW QUESTION 23
- (Exam Topic 1)

You run a local-exec provisioner in a null resource called null_resource.run_script and realize that you need to rerun the script.
Which of the following commands would you use first?

A. terraform taint null_resource.run_script
B. terraform apply -target=null_resource.run_script
C. terraform validate null_resource.run_script
D. terraform plan -target=null_resource.run_script

Answer: A

Explanation:
https://www.terraform.io/cli/commands/taint

NEW QUESTION 24
- (Exam Topic 1)
If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?

A. Run terraform refresh
B. It will happen automatically
C. Manually update the state fire
D. Run terraform import

Answer: A

Explanation:
https://www.terraform.io/cli/commands/refresh#:~:text=The%20terraform%20refresh%20command%20reads%

NEW QUESTION 27
- (Exam Topic 1)
Your DevOps team is currently using the local backend for your Terraform configuration. You would like to
move to a remote backend to begin storing the state file in a central location. Which of the following backends would not work?

A. Amazon S3
B. Artifactory
C. Git
D. Terraform Cloud

Answer: C

Explanation:
https://www.terraform.io/cdktf/concepts/remote-backends https://docs.gitlab.com/ee/user/infrastructure/iac/terraform_state.html

NEW QUESTION 28
- (Exam Topic 1)
How is terraform import run?

A. As a part of terraform init
B. As a part of terraform plan
C. As a part of terraform refresh
D. By an explicit call
E. All of the above

Answer: D

Explanation:
"The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration. Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped. While this may seem tedious, it still gives Terraform users an avenue for importing existing resources."
https://www.terraform.io/cli/import/usage

NEW QUESTION 33
- (Exam Topic 1)
Which backend does the Terraform CLI use by default?

A. Terraform Cloud
B. Consul
C. Remote
D. Local

Answer: D

Explanation:
"By default, Terraform implicitly uses a backend called local to store state as a local file on disk. Every other
backend stores state in a remote service of some kind, which allows multiple people to access it. Accessing state in a remote service generally requires some kind of access credentials, since state data contains extremely sensitive information." https://www.terraform.io/language/settings/backends

NEW QUESTION 35

- (Exam Topic 2)
When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

A. False
B. True

Answer: B

Explanation:
TF_LOG_PATH specifies where the log should persist its output to. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
For example, to always write the log to the directory you're currently running terraform from: export TF_LOG_PATH=./terraform.log
export TF_LOG=TRACE

NEW QUESTION 40
- (Exam Topic 2)
While using generic git repository as a module source, which of the below options allows terraform to select a specific version or tag instead of selecting the HEAD.

A. Append ref argument asmodule "vpc" { source = "git::https://example.com/vpc.git?ref=v1.2.0"}
B. Append version argument asmodule "vpc" { source = "git::https://example.com/vpc.git?version=v1.2.0"}
C. Append ref argument asmodule "vpc" { source = "git::https://example.com/vpc.git#ref=v1.2.0"}
D. By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository and you can not override this.

Answer: A

Explanation:
By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository. You can override this using the ref argument:
module "vpc" {
source = "git::https://example.com/vpc.git?ref=v1.2.0"
}
The value of the ref argument can be any reference that would be accepted by the git checkout command, including branch and tag names.
https://www.terraform.io/docs/modules/sources.html

NEW QUESTION 44
- (Exam Topic 2)
You want to use terraform import to start managing infrastructure that was not originally provisioned through infrastructure as code. Before you can import the resource's current state, what must you do in order to prepare to manage these resources using Terraform?

A. Run terraform refresh to ensure that the state file has the latest information for existing resources.
B. Update the configuration file to include the new resources.
C. Shut down or stop using the resources being imported so no changes are inadvertently missed.
D. Modify the Terraform state file to add the new resources.

Answer: B

Explanation:
The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration.
Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.
The terraform import command is used to import existing infrastructure.
To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.
Example:
resource "aws_instance" "import_example" {
# ...instance configuration...
}
Now terraform import can be run to attach an existing instance to this resource configuration.
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 45
- (Exam Topic 2)
Which one of the following will run echo 0 and echo 1 on a newly created host?

A. provisioner "local-exec" { command = "echo 0" command = "echo 1"}
B. provisioner "remote-exec" { inline = [echo 0,echo 1]}
C. provisioner "remote-exec" {command = "${echo 0}" command = "${echo 1}"}
D. provisioner "remote-exec" { inline = ["echo 0","echo 1"]}

Answer: D

Explanation:
remote-exec Provisioner Example usage
resource "aws_instance" "web" {
...
provisioner "remote-exec" { inline = [

```
"puppet apply",
"consul join ${aws_instance.web.private_ip}",
]
}
}
```

NEW QUESTION 48
- (Exam Topic 2)
The terraform init command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing configuration or state.

A. False
B. True

Answer: B

Explanation:
https://www.terraform.io/docs/commands/init.html

NEW QUESTION 49
- (Exam Topic 2)
What is the default backend for Terraform?

A. consul
B. gcs
C. local
D. etcd

Answer: C

Explanation:
By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to. https://www.terraform.io/docs/backends/index.html

NEW QUESTION 52
- (Exam Topic 2)
You have created a custom variable definition file testing.tfvars. How will you use it for provisioning infrastructure?

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

Answer: C

Explanation:
https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION 57
- (Exam Topic 2)
Which of the following Terraform files should be ignored by Git when committing code to a repo? (select Three)

A. Files named exactly terraform.tfvars or terraform.tfvars.json.
B. Any files with names ending in .auto.tfvars or .auto.tfvars.json.
C. input.tf
D. terraform.tfstate
E. output.tf

Answer: ABD

Explanation:
The .gitignore file should be configured to ignore Terraform files that either contain sensitive data or are not required to save.
Terraform state (terraform.tfstate) can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.
When using local state, state is stored in plain-text JSON files.
The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others.

NEW QUESTION 58
- (Exam Topic 2)
Terraform works well in Windows but a Windows server is required.

A. False
B. True

Answer: A

Explanation:
You may see this QUESTION NO: in actual exam. Please remember : Terraform does not require GO language to be installed as a prerequisite and it does not

require a Windows Server as well.

NEW QUESTION 62
- (Exam Topic 2)
You have declared a variable name my_var in terraform configuration without a value associated with it. variable my_var {}
After running terraform plan it will show an error as variable is not defined.

A. True
B. False

Answer: B

Explanation:
Input variables are usually defined by stating a name, type and a default value. However, the type and default values are not strictly necessary. Terraform can deduct the type of the variable from the default or input value.
Variables can be predetermined in a file or included in the command-line options. As such, the simplest variable is just a name while the type and value are selected based on the input.
variable "variable_name" {}
terraform apply -var variable_name="value"
The input variables, like the one above, use a couple of different types: strings, lists, maps, and boolean. Here are some examples of how each type are defined and used.
String
Strings mark a single value per structure and are commonly used to simplify and make complicated values more user-friendly. Below is an example of a string variable definition.
variable "template" { type = string
default = "01000000-0000-4000-8000-000030080200"
}
A string variable can then be used in resource plans. Surrounded by double quotes, string variables are a simple substitution such as the example underneath.
storage = var.template List
Another type of Terraform variables lists. They work much like a numbered catalogue of values. Each value can be called by their corresponding index in the list.
Here is an example of a list variable definition.
variable "users" { type = list
default = ["root", "user1", "user2"]
}
Lists can be used in the resource plans similarly to strings, but you'll also need to denote the index of the value you are looking for.
username = var.users[0] Map
Maps are a collection of string keys and string values. These can be useful for selecting values based on predefined parameters such as the server configuration by the monthly price.
variable "plans" { type = map default = {
"5USD" = "1xCPU-1GB" "10USD" = "1xCPU-2GB" "20USD" = "2xCPU-4GB"
}
}
You can access the right value by using the matching key. For example, the variable below would set the plan to "1xCPU-1GB".
plan = var.plans["5USD"]
The values matching to their keys can also be used to look up information in other maps. For example, underneath is a shortlist of plans and their corresponding storage sizes.
variable "storage_sizes" { type = map
default = {
"1xCPU-1GB" = "25"
"1xCPU-2GB" = "50"
"2xCPU-4GB" = "80"
}
}
These can then be used to find the right storage size based on the monthly price as defined in the previous example.
size = lookup(var.storage_sizes, var.plans["5USD"])
Boolean
The last of the available variable type is boolean. They give the option to employ simple true or false values. For example, you might wish to have a variable that decides when to generate the root user password on a new deployment.
variable "set_password" { default = false
}
The above example boolean can be used similarly to a string variable by simply marking down the correct variable.
create_password = var.set_password
By default, the value is set to false in this example. However, you can overwrite the variable at deployment by assigning a different value in a command-line variable.
terraform apply -var set_password="true"

NEW QUESTION 65
- (Exam Topic 2)
What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

A. Local backends
B. Providers
C. Remote backends
D. Workspaces

Answer: D

Explanation:
Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure.
Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of

that configuration to be deployed without configuring a new backend or changing authentication credentials.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION 70
- (Exam Topic 2)
Please identify the offerings which are unique to Terraform Enterprise, and not available in either Terraform OSS, or Terraform Cloud. Select four.

A. Audit Logs
B. Private Network Connectivity
C. VCS Integration
D. Sentinel
E. Clustering

Answer: ABE

Explanation:
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 73
- (Exam Topic 2)
When using remote state, state is only ever held in memory when used by Terraform.

A. False
B. True

Answer: B

NEW QUESTION 78
- (Exam Topic 2)
Which of the following best describes a Terraform provider?

A. A plugin that Terraform uses to translate the API interactions with the service or provider.
B. Serves as a parameter for a Terraform module that allows a module to be customized.
C. Describes an infrastructure object, such as a virtual network, compute instance, or other components.
D. A container for multiple resources that are used together.

Answer: A

Explanation:
A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g.Terraform Cloud, DNSimple, Cloudflare).
https://www.terraform.io/docs/providers/index.html

NEW QUESTION 81
- (Exam Topic 2)
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a ______.

A. First Time Configuration
B. Default Configuration
C. Changing Configuration
D. Partial Configuration
E. Incomplete Configuration

Answer: D

Explanation:
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.
With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments:
* Interactively: Terraform will interactively ask you for the required values, unless interactive input is disabled. Terraform will not prompt for optional values.
* File: A configuration file may be specified via the init command line. To specify a file, use the
-backend-config=PATH option when running terraform init. If the file contains secrets it may be kept in a secure data store, such as Vault, in which case it must be downloaded to the local disk before running Terraform.
* Command-line key/value pairs: Key/value pairs can be specified via the init command line. Note that many shells retain command-line flags in a history file, so this isn't recommended for secrets. To specify a single key/value pair, use the -backend-config="KEY=VALUE" option when running terraform init.
https://www.terraform.io/docs/backends/config.html#partial-configuration

NEW QUESTION 83
- (Exam Topic 2)
Which of the following best describes the default local backend?

A. The local backend is where Terraform Enterprise stores logs to be processed by an log collector.
B. The local backend stores state on the local filesystem, locks the state using system APIs, and performs operations locally.
C. The local backend is the directory where resources deployed by Terraform have direct access to in order to update their current state.
D. The local backend is how Terraform connects to public cloud services, such as AWS, Azure, or GCP.

Answer: B

Explanation:
The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.
terraform { backend "local" {
path = "relative/path/to/terraform.tfstate"
}
}
https://www.terraform.io/docs/backends/types/local.html

NEW QUESTION 87
- (Exam Topic 2)
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.

Answer: AC

Explanation:
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

NEW QUESTION 90
- (Exam Topic 2)
What does terrafom plan do ?

A. Create an execution plan by evaluating the difference between configuration file and state file.
B. Performs a refresh, unless explicitly disabled, and then apply the changes that are necessary to achieve the desired state specified in the configuration files.
C. Create an execution plan by evaluating the difference between configuration file and actual infrastructure.
D. Checks whether the execution plan for a set of changes matches your expectations by making changes to real resources or to the state.

Answer: A

NEW QUESTION 94
- (Exam Topic 3)
Which of the below options is the equivalent Terraform 0.12 version of the snippet which is written in Terraform 0.11?
"${var.instance_id}"

A. variable.instance_id
B. var.instance_ids
C. var.instance_id
D. None of the above

Answer: C

NEW QUESTION 97
- (Exam Topic 3)
You have been given requirements to create a security group for a new application. Since your organization standardizes on Terraform, you want to add this new security group with the fewest number of lines of code. What feature could you use to iterate over a list of required tcp ports to add to the new security group?

A. dynamic backend
B. splat expression
C. terraform import
D. dynamic block

Answer: D

Explanation:
A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value and generates a nested block for each element of that complex value.
https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks

NEW QUESTION 99
- (Exam Topic 3)
Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?

A. False
B. True

Answer: A

Explanation:
Terraform module need not be public and open-source. Module can be placed in
* Local paths
* Terraform Registry
* GitHub
* Bitbucket
* Generic Git, Mercurial repositories
* HTTP URLs
* S3 buckets
* GCS buckets https://www.terraform.io/docs/modules/sources.html

NEW QUESTION 103
- (Exam Topic 3)
Which of the following challenges would Terraform be a candidate for solving? (Select THREE)

A. Enable self-service infrastructure to allocate resources on your proprietary private cloud.
B. Reduce the number of workflows needed for managing infrastructure across each of the companies public and private clouds.
C. Utilize a single tool for all of the infrastructure and configuration management needs.
D. Have a single interoperable tool to manage the variety of services including GitHub repositories, MySQL database, and Kubernetes clusters.

Answer: ABD

NEW QUESTION 106
- (Exam Topic 3)
The canonical format may change in minor ways between Terraform versions, so after upgrading Terraform it is recommended to proactively run.

A. terraform fmt
B. terraform init
C. terraform validate
D. terraform plan

Answer: A

NEW QUESTION 109
- (Exam Topic 3)
Which of the following value will be accepted for var1? variable "var1" {
type = string
}

A. None of the above
B. Both A and B
C. "5"
D. 5

Answer: B

Explanation:
Terraform automatically converts number and bool values to strings when needed.

NEW QUESTION 114
- (Exam Topic 3)
Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.

A. Create individual workspaces for each team , and ask them to share modules across workspaces.
B. Implement a Private module registry in Terraform cloud , and ask teams to reference them.
C. Upgrade to Terraform enterprise , since this is not possible in terraform cloud.
D. Upload the modules in the terraform public module registry , and ask teams to reference them

Answer: B

Explanation:
Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster.
By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar.
Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise. Terraform Cloud's private module registry helps you share Terraform modules across your organization.
https://www.terraform.io/docs/cloud/registry/index.html https://www.terraform.io/docs/cloud/registry/publish.html

NEW QUESTION 117
- (Exam Topic 3)

* 1. resource "aws_s3_bucket" "example" {
* 2. bucket = "my-test-s3-terraform-bucket"
* 3. …} resource "aws_iam_role" "test_role" {
* 4. name = "test_role"
* 5. …}

Due to the way that the application code is written, the s3 bucket must be created before the test role is created, otherwise there will be a problem. How can you ensure that?

A. Add explicit dependency using depends_on . This will ensure the correct order of resource creation.
B. This will already be taken care of by terraform native implicit dependenc
C. Nothing else needs to be done from your end.
D. This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.
E. Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.

Answer: A

Explanation:
Implicit dependency works only if there is some reference of one resource to another. Explicit dependency is the option here.

NEW QUESTION 118
- (Exam Topic 3)
Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.

A. terraform state
B. terraform apply
C. terraform plan
D. terraform validate
E. terraform output

Answer: BC

NEW QUESTION 120
- (Exam Topic 3)
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

Answer: B

Explanation:
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

NEW QUESTION 124
- (Exam Topic 3)
When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?

A. Object Storage
B. State Locking
C. WorkSpaces
D. Encryption

Answer: B

Explanation:
If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state.
State locking happens automatically on all operations that could write state. 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.
Not all backends support locking. Please view the list of backend types for details on whether a backend supports locking or not.
https://www.terraform.io/docs/state/locking.html

NEW QUESTION 126
- (Exam Topic 3)
Which of the below commands will rename a EC2 instance without destroying and recreating it?

A. terraform state mv
B. terraform mv
C. terraform plan
D. terraform plan mv

Answer: A

NEW QUESTION 128
- (Exam Topic 3)
Which of the below datatype is not supported by Terraform.

A. Array
B. List
C. Object
D. Map

Answer: A

NEW QUESTION 133
- (Exam Topic 3)
A data block requests that Terraform read from a given data source and export the result under the given local name.

A. False
B. True

Answer: B

NEW QUESTION 136
- (Exam Topic 3)
You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.

A. Run the command export TF_LOG_FILE=./terraform.log.
B. Run the command export TF_LOG_PATH=./terraform.log.
C. Run the command export TF_DEBUG_PATH=./terraform.log.
D. No explicit action require
E. Terraform will take care of this as you have enable TF_LOG.

Answer: B

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

NEW QUESTION 137
- (Exam Topic 3)
Command terraform refresh will update state file?

A. False
B. True

Answer: B

Explanation:
The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. This can be used to detect any drift from the last-known state, and to update the state file.
This does not modify infrastructure, but does modify the state file. If the state is changed, this may cause changes to occur during the next plan or apply.
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION 142
- (Exam Topic 3)
The terraform state command can be used to _____

A. Update current state
B. Refresh existing state file
C. Print the current state file in console
D. It is not a valid command

Answer: A

Explanation:
The terraform state command is used for advanced state management. Rather than modify the state directly, the terraform state commands can be used in many cases instead.
https://www.terraform.io/docs/commands/state/index.html

NEW QUESTION 145
- (Exam Topic 3)
Terraform-specific settings and behaviors are declared in which configuration block type?

A. provider
B. terraform
C. resource
D. data

Answer: B

Explanation:
The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.
Example terraform {
required_version = "> 0.12.0"
}
https://www.terraform.io/docs/configuration/terraform.html

NEW QUESTION 147
- (Exam Topic 3)
A single terraform resource file that defines an aws_instance resource can simply be renamed to vsphere_virtual_machine in order to switch cloud providers.

A. True
B. False

Answer: B

Explanation:
Every provider has its own required and allowed declarations none of which match between cloud providers.

NEW QUESTION 150
- (Exam Topic 3)
During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

A. The resource will be planned for destruction and recreation upon the next terraform apply
B. Terraform will retry to provision again.
C. The failure of provisioner will be ignored and it will not cause a failure to terraform apply
D. The resource will be automatically destroyed.

Answer: A

Explanation:
If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply. Terraform does this because a failed provisioner can leave a resource in a semi-configured state. Because Terraform cannot reason about what the provisioner does, the only way to ensure proper creation of a resource is to recreate it. This is tainting.
You can change this behavior by setting the on_failure attribute, which is covered in detail below. https://www.terraform.io/docs/provisioners/index.html#creation-time-provisioners https://www.terraform.io/docs/provisioners/index.html#destroy-time-provisioners https://www.terraform.io/docs/provisioners/index.html#failure-behavior

NEW QUESTION 151
- (Exam Topic 4)
Using the terraform state rm command against a resource will destroy it.

A. True
B. False

Answer: B

NEW QUESTION 152
- (Exam Topic 4)
Valarie has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code. Valarie wants to hide the output value in the CLI after terraform apply that's why she has used sensitive parameter.
* 1. output "db_password" {
* 2. value = local.db_password
* 3. sensitive = true
* 4. }
Since sensitive is set to true, will the value associated with db password be available in plain-text in the state file for everyone to read?

A. Yes
B. No

Answer: A

Explanation:
Outputs can be marked as containing sensitive material by setting the sensitive attribute to true, like this: output "sensitive" {
sensitive = true value = VALUE
}
When outputs are displayed on-screen following a terraform apply or terraform refresh, sensitive outputs are redacted, with <sensitive> displayed in place of their value.
Limitations of Sensitive Outputs
The values of sensitive outputs are still stored in the Terraform state, and available using the terraform output command, so cannot be relied on as a sole means of protecting values.
Sensitivity is not tracked internally, so if the output is interpolated in another module into a resource, the value will be displayed.

NEW QUESTION 153

- (Exam Topic 4)
Select the answer below that completes the following statement: Terraform Cloud can be managed from the CLI but requires ______?

A. an API token
B. a TOTP token
C. a username and password
D. authentication using MFA

Answer: A

Explanation:
API and CLI access are managed with API tokens, which can be generated in the Terraform Cloud UI. Each user can generate any number of personal API tokens, which allow access with their own identity and permissions. Organizations and teams can also generate tokens for automating tasks that aren't tied to an individual user.

NEW QUESTION 155
- (Exam Topic 4)
As a developer, you want to ensure your plugins are up to date with the latest versions. Which Terraform command should you use?

A. terreform providers- upgrade
B. terreform apply -upgrade
C. terreform refresh -upgrade
D. terreformn lnit -upgrade

Answer: D

NEW QUESTION 158
- (Exam Topic 4)
What does the command terraform fmt do?

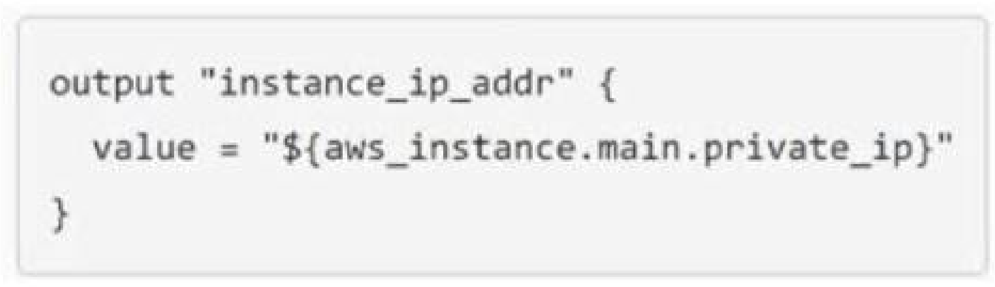
A. Rewrite Terraform configuration files to a canonical format and style.
B. Deletes the existing configuration file.
C. Updates the font of the configuration file to the official font supported by HashiCorp.
D. Formats the state file in order to ensure the latest state of resources can be obtained.

Answer: A

Explanation:
The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style. This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability.
Other Terraform commands that generate Terraform configuration will produce configuration files that conform to the style imposed by terraform fmt, so using this style in your own files will ensure consistency.
https://www.terraform.io/docs/commands/fmt.html

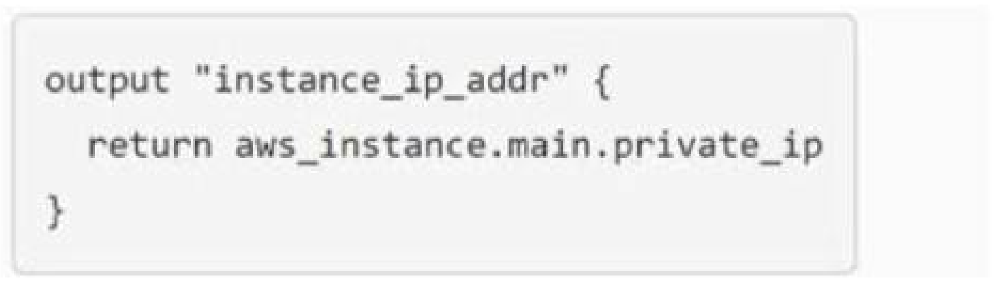
NEW QUESTION 161
- (Exam Topic 4)
You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name the instance resource "main".
Which of these is the correct way to define the output value using HCL2?

A.



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

B.

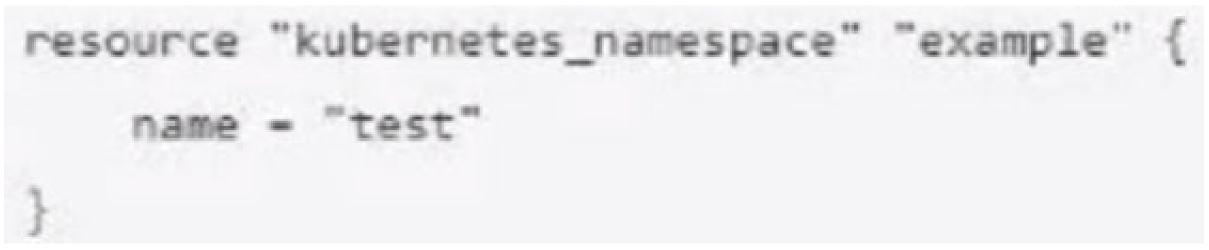


```
output "instance_ip_addr" {
  return aws_instance.main.private_ip
}
```

A. Option A
B. Option B

Answer: A

NEW QUESTION 163
- (Exam Topic 4)
How would you reference the attribute "name* of this fictitious resource in HCL?



```
resource "kubernetes_namespace" "example" {
    name = "test"
}
```

A. resource.kubrnetes_namespace>example.name
B. kubernetes_namespace.test.name
C. kubernetes_namespace.example,name
D. data kubernetes_namespace.name
E. None of the above

Answer: C

Explanation:
https://www.terraform.io/language/expressions/references#references-to-resource-attributes

NEW QUESTION 166
- (Exam Topic 4)
Which of the following actions are performed during a terraform init?

A. Initializes downloaded and/or installed providers
B. Initializes the backend configuration
C. Provisions the declared resources in your configuration
D. Download the declared providers which are supported by HashiCorp

Answer: ABD

Explanation:
The terraform init command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times.
This command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing
configuration or state. terraform init command does * Copy a Source Module
* Backend Initialization
* Child Module Installation
* Plugin Installation https://www.terraform.io/docs/commands/init.html

NEW QUESTION 170
- (Exam Topic 4)
Which of the following can you do with terraform plan? Choose two correct answers.

A. View the execution plan and check if the changes match your expectations
B. Schedule Terraform to run at a planned time in the future
C. Execute a plan in a different workspace
D. Save a generated execution plan to apply later

Answer: AD

Explanation:
https://learn.hashicorp.com/tutorials/terraform/plan

NEW QUESTION 174
- (Exam Topic 4)
Which of the following statements about Terraform modules is not true?

A. Modules must be publicly accessible
B. Modules can be called multiple times
C. Module is a container for one or more resources
D. Modules can call other modules

Answer: A

Explanation:
In addition to modules from the local filesystem, Terraform can load modules from a public or private registry. Also, members of your organization might produce modules specifically crafted for your own infrastructure needs. Source: https://www.terraform.io/language/modules

NEW QUESTION 176
- (Exam Topic 4)
Why should secrets not be hard coded into Terraform code? Choose two correct answers

A. All passwords should be rotated on a quarterly basis.
B. The Terraform code is copied to the target resources to be applied locally and could expose secrets if a target resource is compromised.
C. Terraform code is typically stored in version control, as well as copied to the systems from h it's run.Any of those may not have robust security mechanisms.
D. It makes the code less reusable.

Answer: BC

NEW QUESTION 177
- (Exam Topic 4)
Which of the following arguments are required when declaring a Terraform output?

A. sensitive
B. description
C. default
D. value

Answer: D

NEW QUESTION 182
- (Exam Topic 4)
Which of the following is not a benefit of adopting infrastructure as code?

A. Automation
B. Versioning
C. Reusability of code
D. Interpolation

Answer: D

NEW QUESTION 186
- (Exam Topic 4)
Running terraform fmt without any flags in a directory with Terraform configuration files will check the formatting of those files without changing their contents.

A. True
B. False

Answer: B

Explanation:
The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style.

NEW QUESTION 190
- (Exam Topic 4)
Which of the following statements best describes the Terraform list(...) type?

A. a collection of values where each is identified by a string label.
B. a sequence of values identified by consecutive whole numbers starting with zero.
C. a collection of unique values that do not have any secondary identifiers or ordering.
D. a collection of named attributes that each have their own type.

Answer: B

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

NEW QUESTION 195
- (Exam Topic 4)
When using providers that require the retrieval of data, such as the HashiCorp Vault provider, in what phase does Terraform actually retrieve the data required?

A. terraform delete
B. terraform plan
C. terraform init
D. terraform apply

Answer: C

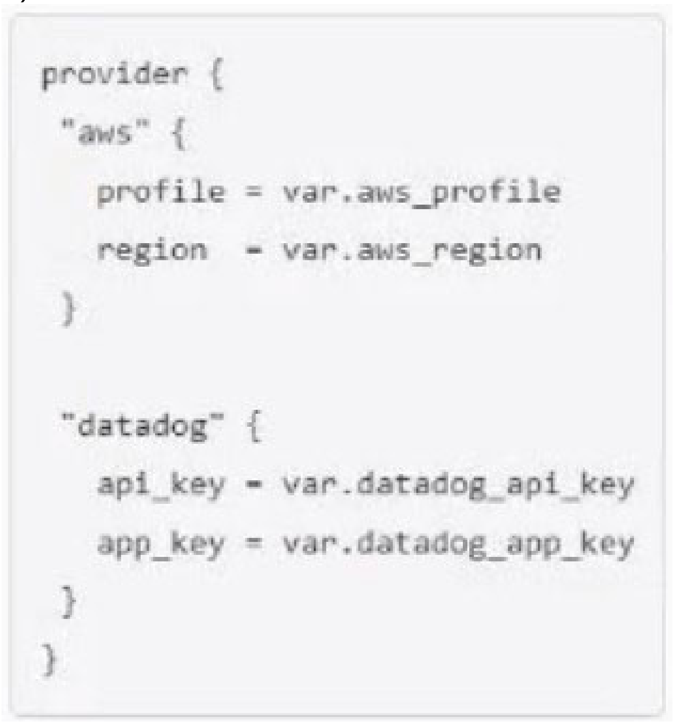
NEW QUESTION 199
- (Exam Topic 4)
Terraform variable names are saved in the state file.

A. True
B. False

Answer: B

Explanation:
Terraform stores information about your infrastructure in a state file. This state file keeps track of resources created by your configuration and maps them to real-world resources. https://learn.hashicorp.com/tutorials/terraform/state-cli

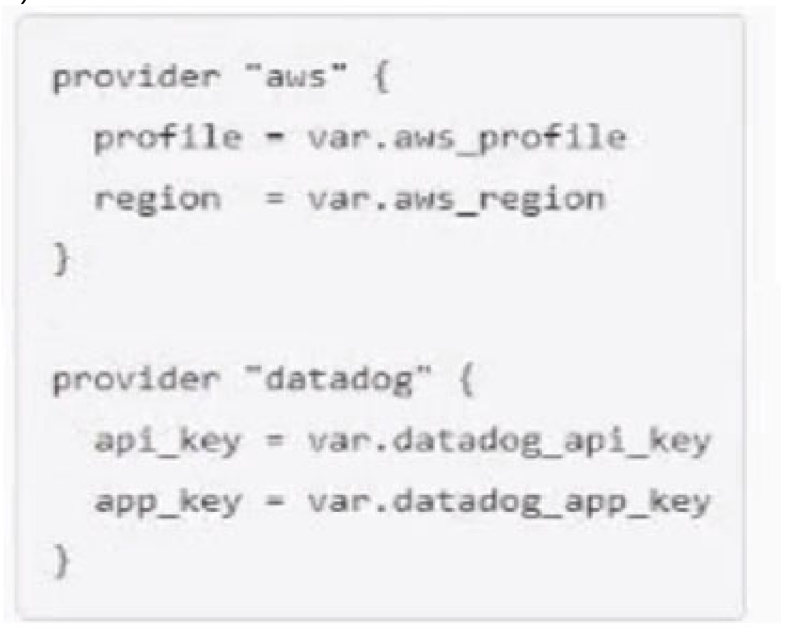
NEW QUESTION 202
- (Exam Topic 4)
You are creating a Terraform configuration which needs to make use of multiple providers, one for AWS and one for Datadog.
Which of the following provider blocks would allow you to do this?
A)



```
provider {
 "aws" {
   profile = var.aws_profile
   region  = var.aws_region
 }

 "datadog" {
   api_key = var.datadog_api_key
   app_key = var.datadog_app_key
 }
}
```

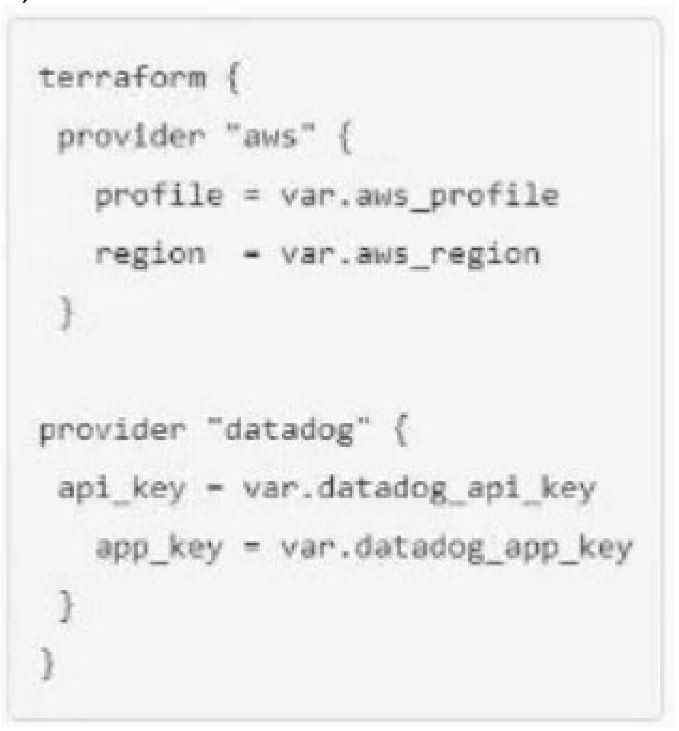
B)



```
provider "aws" {
  profile = var.aws_profile
  region  = var.aws_region
}

provider "datadog" {
  api_key = var.datadog_api_key
  app_key = var.datadog_app_key
}
```

C)



```
terraform {
 provider "aws" {
   profile = var.aws_profile
   region  = var.aws_region
 }

provider "datadog" {
 api_key = var.datadog_api_key
   app_key = var.datadog_app_key
 }
}
```

A. Option A
B. Option B
C. Option C

Answer: B

Explanation:
https://www.terraform.io/language/providers/configuration

NEW QUESTION 203
- (Exam Topic 4)
State is a requirement for Terraform to function

A. True
B. False

Answer: A

Explanation:
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run.
Purpose of Terraform State
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.
As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).
* 1. Mapping to the Real World
Terraform requires some sort of database to map Terraform config to the real world. When you have a resource resource "aws_instance" "foo" in your configuration, Terraform uses this map to know that instance i- abcd1234 is represented by that resource.
For some providers like AWS, Terraform could theoretically use something like AWS tags. Early prototypes of Terraform actually had no state files and used this method. However, we quickly ran into problems. The first major issue was a simple one: not all resources support tags, and not all cloud providers support tags.
Therefore, for mapping configuration to resources in the real world, Terraform uses its own state structure.
* 2. Metadata
Alongside the mappings between resources and remote objects, Terraform must also track metadata such as resource dependencies.
Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.
To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.
One way to avoid this would be for Terraform to know a required ordering between resource types. For example, Terraform could know that servers must be deleted before the subnets they are a part of. The
complexity for this approach quickly explodes, however: in addition to Terraform having to understand the ordering semantics of every resource for every cloud, Terraform must also understand the ordering across providers.
Terraform also stores other metadata for similar reasons, such as a pointer to the provider configuration that was most recently used with the resource in situations where multiple aliased providers are present.
* 3. Performance
In addition to basic mapping, Terraform stores a cache of the attribute values for all resources in the state. This is the most optional feature of Terraform state and is done only as a performance improvement.
When running a terraform plan, Terraform must know the current state of resources in order to effectively determine the changes that it needs to make to reach your desired configuration.
For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.
For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
* 4. Syncing
In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but when using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects.
Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION 206
- (Exam Topic 4)
You decide to move a Terraform state file to Amazon S3 from another location. You write the code below into a file called\

```
terraform {
  backend "s3" {
    bucket - "my-tf-bucket"
    region = "us-east-1"
  }
}
```

You immediately run terraform apply but don't see any changes. Your state file didn't move. Which command
will migrate your current state file to the new S3 remote backend?

A. terraform push
B. terraform init
C. terraform refresh
D. terraform state

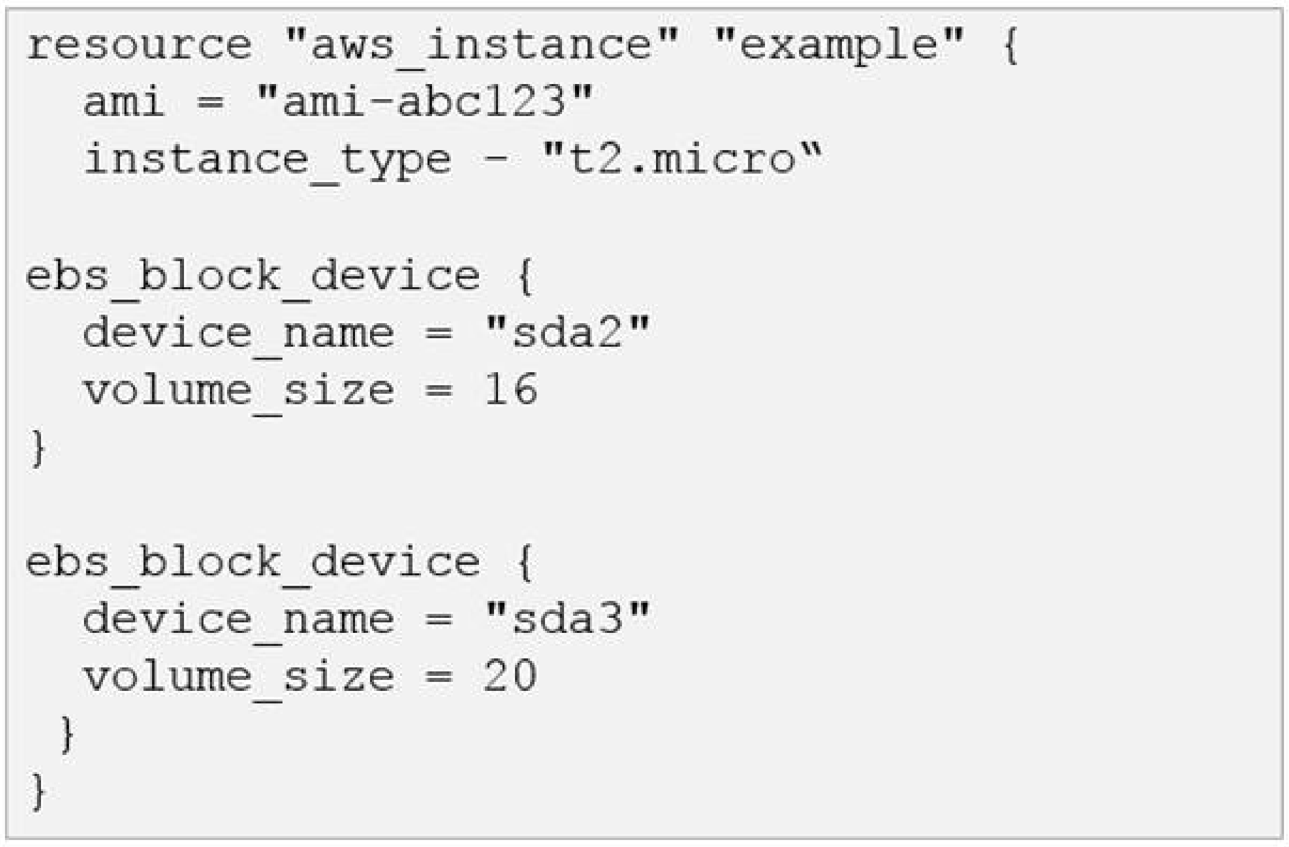
Answer: B

NEW QUESTION 207
- (Exam Topic 4)
colleagues is new toTerraform and wants to add a new workspace named new-hire. What command he should execute from the following?

A. terraform workspace-new-new-hire
B. terraform workspace new new hire
C. terraform workspace init new-hire
D. terraform workspace new-hire

Answer: B

NEW QUESTION 208
- (Exam Topic 4)
How would you reference the Volume IDs associated with the ebs_block_device blocks in this configuration?



```
resource "aws_instance" "example" {
  ami = "ami-abc123"
  instance_type - "t2.micro"

ebs_block_device {
  device_name = "sda2"
  volume_size = 16
}

ebs_block_device {
  device_name = "sda3"
  volume_size = 20
 }
}
```

A. aws_instance.example.ebs_block_device.[*].volume_id
B. aws_instance.example.ebs_block_device.volume_id
C. aws_instance.example.ebs_block_device[sda2,sda3].volume_id
D. aws_instance.example.ebs_block_device.*.volume_id

Answer: A

Explanation:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/device_naming.html

NEW QUESTION 210
- (Exam Topic 4)
What does terraform import allow you to do?

A. Import a new Terraform module
B. Use a state file to import infrastructure to the cloud
C. Import provisioned infrastructure to your state file
D. Import an existing state file to a new Terraform workspace

Answer: C

NEW QUESTION 211
- (Exam Topic 4)
Module version is required to reference a module on the Terraform Module Registry.

A. True
B. False

Answer: B

NEW QUESTION 216
- (Exam Topic 4)
How does Terraform determine dependencies between resources?

A. Terraform automatically builds a resource graph based on resources, provisioners, special meta-parameters, and the state file, if present.
B. Terraform requires all dependencies between resources to be specified using the depends_on parameter
C. Terraform requires resources in a configuration to be listed in the order they will be created to determine dependencies
D. Terraform requires resource dependencies to be defined as modules and sourced in order

Answer: A

Explanation:
https://learn.hashicorp.com/tutorials/terraform/dependencies

NEW QUESTION 219
- (Exam Topic 4)
Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.
What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

A. Submit a ticket to AWS and ask them to export the state of all existing resources and use terraform import to import them into the state file.
B. Delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward.
C. Resources that are manually deployed in the AWS console cannot be imported by Terraform.
D. Using terraform import, import the existing infrastructure into your Terraform state.

Answer: D

Explanation:
Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.
This is a great way to slowly transition infrastructure to Terraform.
The terraform import command is used to import existing infrastructure.
To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.
Example:

```
resource "aws_instance" "import_example" {
# ...instance configuration...
}
```

Now terraform import can be run to attach an existing instance to this resource configuration.

```
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
```

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside
Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 220
- (Exam Topic 4)
Which of the following is not supported backend types in Terra form?

A. consul
B. gcs
C. manta
D. bitbucket

Answer: D

NEW QUESTION 223
- (Exam Topic 4)
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

Answer: ACDE

Explanation:

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

NEW QUESTION 228
- (Exam Topic 4)
Jack is a newbieto Terraform and wants to enable detailed logging to find all the details. Which environment variable does he need to set?

A. TF_help
B. TF LOG
C. TF_Debug
D. TF_var_log

Answer: B

NEW QUESTION 232
- (Exam Topic 4)
While Terraform is generally written using the HashiCorp Configuration Language (HCL), what other syntax can Terraform are expressed in?

A. JSON
B. YAML

C. TypeScript
D. XML

Answer: A

Explanation:
The constructs in the Terraform language can also be expressed in JSON syntax, which is harder for humans to read and edit but easier to generate and parse programmatically.

NEW QUESTION 235
- (Exam Topic 4)
Which feature of Terraform allows multiple state files for a single configuration file depending upon the environment?

A. Terraform Modules
B. Terraform Enterprise
C. Terraform Workspaces
D. Terraform Remote Backends

Answer: C

NEW QUESTION 237
- (Exam Topic 4)
Named workspaces are not a suitable isolation mechanism for strong separation between staging and production?

A. True
B. False

Answer: A

Explanation:
Organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g. staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are not a suitable isolation mechanism for this scenario.
https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces

NEW QUESTION 240
- (Exam Topic 4)
Changing the Terraform backend from the default "local" backend to a different one after doing your first terraform apply is:

A. Mandatory
B. Optional
C. Impossible
D. Discouraged

Answer: B

NEW QUESTION 245
- (Exam Topic 4)
Select all Operating Systems that Terraform is available for. (select five)

A. Linux
B. macOS
C. Unix
D. Solaris
E. Windows
F. FreeBSD

Answer: ABDEF

Explanation:

Terraform is available for macOS, FreeBSD, OpenBSD, Linux, Solaris, Windows https://www.terraform.io/downloads.html

NEW QUESTION 247
- (Exam Topic 4)
What kind of configuration block will create an infrastructure object with settings specified in the block?

A. state
B. provider
C. resource
D. data

Answer: C

NEW QUESTION 252
- (Exam Topic 4)

A "backend" in Terraform determines how state is loaded and how an operation such as apply is executed. Which of the following is not a supported backend type?

A. Terraform enterprise
B. Consul
C. Github
D. S3
E. Artifactory

Answer: C

Explanation:
Github is not a supported backend type. https://www.terraform.io/docs/backends/types/index.html

NEW QUESTION 253
- (Exam Topic 4)
You've used Terraform to deploy a virtual machine and a database. You want to replace this virtual machine instance with an identical one without affecting the database. What is the best way to achieve this using Terraform?

A. Use the Terraform taint command targeting the VMs then run Terraform plan and Terraform apply
B. Delete the Terraform VM resources from your Terraform code then run Terraform plan and terraform apply
C. Use the terraform apply command targeting the VM resources only
D. Use the terraform state rm command to remove the VM from state file

Answer: A

Explanation:
https://www.terraform.io/cli/state/taint

NEW QUESTION 255
- (Exam Topic 4)
You cannot install third party plugins using terraform init.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/cli/commands/init
For providers that are published in either the public Terraform Registry or in a third-party provider registry, terraform init will automatically find, download, and install the necessary provider plugins.

NEW QUESTION 257
- (Exam Topic 4)
After executing a terraform apply, you notice that a resource has a tilde (~) next to it. What does this infer?

A. The resource will be updated in place.
B. The resource will be created.
C. Terraform can't determine how to proceed due to a problem with the state file.
D. The resource will be destroyed and recreated.

Answer: A

Explanation:
The prefix -/+ means that Terraform will destroy and recreate the resource, rather than updating it in-place. The prefix ~ means that some attributes and resources can be updated in-place.

```
$ terraform apply
aws_instance.example: Refreshing state... [id=i-0bbf06244e44211d1] An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement Terraform will perform the following actions:
# aws_instance.example must be replaced
-/+ resource "aws_instance" "example" {
~ ami = "ami-2757f631" -> "ami-b374d5a5" # forces replacement
~ arn = "arn:aws:ec2:us-east-1:130490850807:instance/i-0bbf06244e44211d1" -> (known after apply)
~ associate_public_ip_address = true -> (known after apply)
~ availability_zone = "us-east-1c" -> (known after apply)
~ cpu_core_count = 1 -> (known after apply)
~ cpu_threads_per_core = 1 -> (known after apply)
- disable_api_termination = false -> null
- ebs_optimized = false -> null get_password_data = false
+ host_id = (known after apply)
~ id = "i-0bbf06244e44211d1" -> (known after apply)
~ instance_state = "running" -> (known after apply) instance_type = "t2.micro"
~ ipv6_address_count = 0 -> (known after apply)
~ ipv6_addresses = [] -> (known after apply)
+ key_name = (known after apply)
- monitoring = false -> null
+ network_interface_id = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
~ primary_network_interface_id = "eni-0f1ce5bdae258b015" -> (known after apply)
~ private_dns = "ip-172-31-61-141.ec2.internal" -> (known after apply)
~ private_ip = "172.31.61.141" -> (known after apply)
~ public_dns = "ec2-54-166-19-244.compute-1.amazonaws.com" -> (known after apply)
~ public_ip = "54.166.19.244" -> (known after apply)
~ security_groups = [
- "default",
] -> (known after apply) source_dest_check = true
~ subnet_id = "subnet-1facdf35" -> (known after apply)
~ tenancy = "default" -> (known after apply)
~ volume_tags = {} -> (known after apply)
~ vpc_security_group_ids = [
- "sg-5255f429",
] -> (known after apply)
- credit_specification {
- cpu_credits = "standard" -> null
}
+ ebs_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ snapshot_id = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
+ ephemeral_block_device {
+ device_name = (known after apply)
+ no_device = (known after apply)
+ virtual_name = (known after apply)
}
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
~ root_block_device {
~ delete_on_termination = true -> (known after apply)
~ iops = 100 -> (known after apply)
~ volume_id = "vol-0079e485d9e28a8e5" -> (known after apply)
~ volume_size = 8 -> (known after apply)
~ volume_type = "gp2" -> (known after apply)
}
}
Plan: 1 to add, 0 to change, 1 to destroy.
```

NEW QUESTION 258
- (Exam Topic 4)
Terraform plan updates your state file.

A. True
B. False

Answer: B

Explanation:
The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review. Source: https://www.terraform.io/cli/commands/plan

NEW QUESTION 263
- (Exam Topic 4)
Suppose terraformcode is taking up some values which are not defined inside the code files. In which of the following options issue might have occurred?

A. Issue in main.tf file
B. Issue in vars.tf file
C. Issue in terraform.tfvars
D. Issue in Environment Variables

Answer: D

NEW QUESTION 266
- (Exam Topic 4)
You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you terraform apply again immediately afterwards without changing any Terraform code?

A. Terraform will terminate and recreate the VM
B. Terraform will create another duplicate VM

C. Terraform will apply the VM to the state file
D. Nothing

Answer: D

NEW QUESTION 267
- (Exam Topic 4)
During a terraform plan, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

A. Terraform attempts to provision the resource up to three times before exiting with an error
B. the terraform plan is rolled back and all provisioned resources are removed
C. it is automatically deleted
D. the resource is marked as tainted

Answer: D

Explanation:
If a resource successfully creates but fails during provisioning, Terraform will error and mark the resource as "tainted". A resource that is tainted has been physically created, but can't be considered safe to use since provisioning failed. Terraform also does not automatically roll back and destroy the resource during the apply when the failure happens, because that would go against the execution plan: the execution plan would've said a resource will be created, but does not say it will ever be deleted.

NEW QUESTION 271
- (Exam Topic 4)
What is a downside to using the Vault provider to read secrets from Vault?

A. Secrets are persisted to the state file and plans.
B. Terraform and Vault must be running on the same version.
C. Terraform and Vault must be running on the same physical host.
D. Terraform requires a unique auth method to work with Vault.

Answer: A

Explanation:
The Vault provider allows Terraform to read from, write to, and configure Hashicorp Vault.
Interacting with Vault from Terraform causes any secrets that you read and write to be persisted in both Terraform's state file and in any generated plan files. For any Terraform module that reads or writes Vault secrets, these files should be treated as sensitive and protected accordingly.

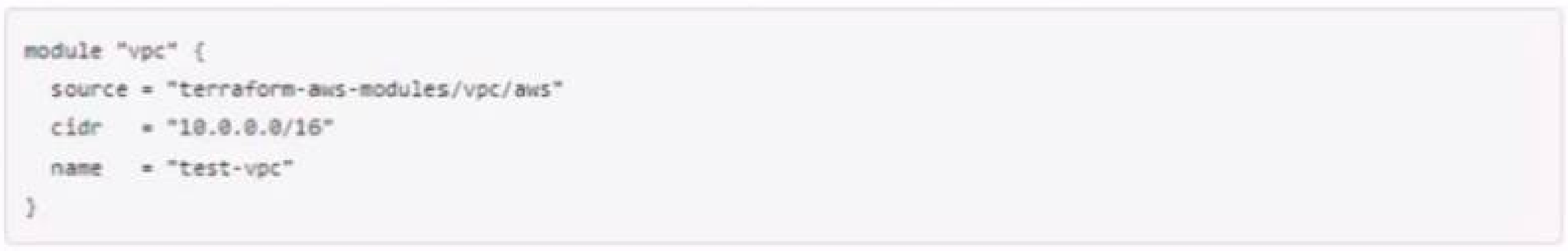
NEW QUESTION 273
- (Exam Topic 4)
When do you need to explicitly execute terraform refresh?

A. Before every terraform plan
B. Before every terraform apply
C. Before every terraform import
D. None of the above

Answer: D

Explanation:
Wherever possible, avoid using terraform refresh explicitly and instead rely on Terraform's behavior of automatically refreshing existing objects as part of creating a normal plan. Source: https://www.terraform.io/cli/commands/refresh

NEW QUESTION 275
- (Exam Topic 4)
In the below configuration, how would you reference the module output vpc_id ?



```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"
  cidr   = "10.0.0.0/16"
  name   = "test-vpc"
}
```

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

A. Mastered
B. Not Mastered

Answer: A

Explanation:
module.vpc.id

NEW QUESTION 277
- (Exam Topic 4)
You need to specify a dependency manually. What resource meta-parameter can you use lo make sure Terraform respects thee dependency?

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

A. Mastered
B. Not Mastered

Answer: A

Explanation:
depends_on

NEW QUESTION 279
- (Exam Topic 4)
When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)

A. directly querying HashiCorp Vault for the secrets
B. command-line key/value pairs
C. use the -backend-config=PATH to specify a separate config file
D. interactively on the command line

Answer: BCD

Explanation:
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.
With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments: https://www.terraform.io/docs/backends/init.html#backend-initialization

NEW QUESTION 282
- (Exam Topic 4)
When you use a remote backend that needs authentication. HashrCorp recommends that you:

A. Push your Tefraform configuration to an encrypted git repository
B. Write the authentication credentials in the Terraform configuration files
C. Use partial configuration to load the authentication credentials outside of the Terraform code
D. Keep the Terraform configuration files in a secret store

Answer: C

Explanation:
We recommend omitting the token from the configuration, and instead using terraform login or manually configuring credentials in the CLI config file. Reference: https://www.terraform.io/language/settings/backends/remote

NEW QUESTION 284
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

Answer: A

NEW QUESTION 285
- (Exam Topic 4)
A variable az has the following default value. What will be the datatype of the variable? az=["us-west-1a","us-east-1a"]

A. Object
B. List
C. Map
D. String

Answer: B

NEW QUESTION 288
- (Exam Topic 4)
Your team lead does not trust the junior terraform engineers who now have access to the git repo . So , he wants you to have some sort of a checking layer , whereby , you can ensure that the juniors will not create any non-compliant resources that might lead to a security audit failure in future. What can you do to efficiently enforce this?

A. Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.
B. Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and writePolicy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.
C. Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using Policy-As-Code.
D. Create a git master branch , and implement PR . Every change needs to be reviewed by you , before being merged to the master branch.

Answer:

B

Explanation:
Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION 292
- (Exam Topic 4)
Terraform Enterprise (also referred to as pTFE) requires what type of backend database for a clustered deployment?

A. PostgreSQL
B. Cassandra
C. MySQL
D. MSSQL

Answer: A

Explanation:

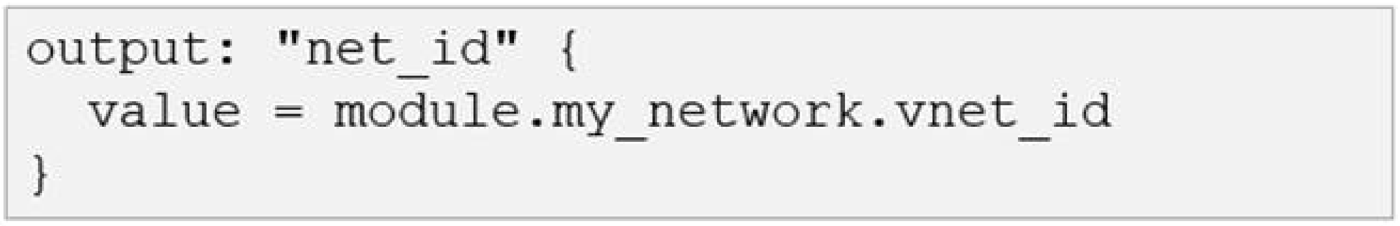
External Services mode stores the majority of the stateful data used by the instance in an external PostgreSQL database and an external S3-compatible endpoint or Azure blob storage. There is still critical data stored on the instance that must be managed with snapshots. Be sure to check the PostgreSQL Requirements for information that needs to be present for Terraform Enterprise to work. This option is best for users with expertise managing PostgreSQL or users that have access to managed PostgreSQL offerings like AWS RDS.

NEW QUESTION 295
- (Exam Topic 4)
Which of the following is not valid source path for specifying a module?

A. source = "./modulelversion=v1.0.0"
B. source = "github.com/hashicorp/example?ref=v1.0.0"
C. source = "./module"
D. source = "hashicorp/consul/aws"

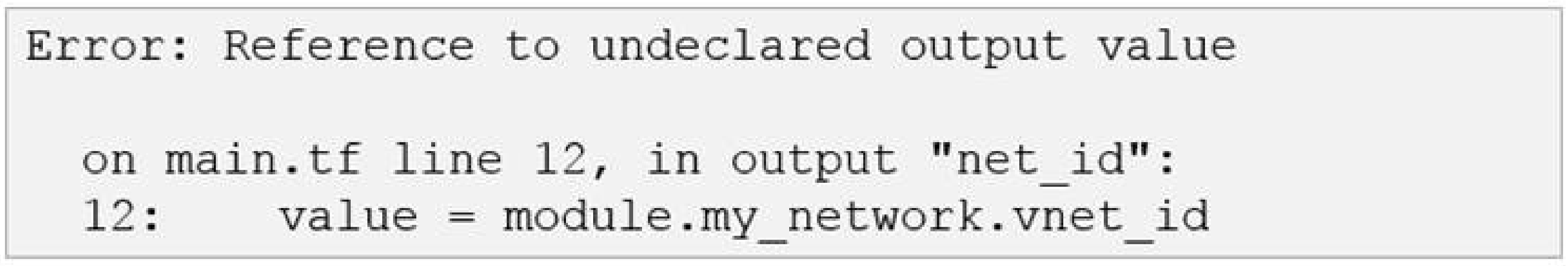
Answer: A

NEW QUESTION 297
- (Exam Topic 4)
You are using a networking module in your Terraform configuration with the name label my_network. In your main configuration you have the following code:



```
output: "net_id" {
  value = module.my_network.vnet_id
}
```

When you run terraform validate, you get the following error:



```
Error: Reference to undeclared output value

  on main.tf line 12, in output "net_id":
  12:     value = module.my_network.vnet_id
```

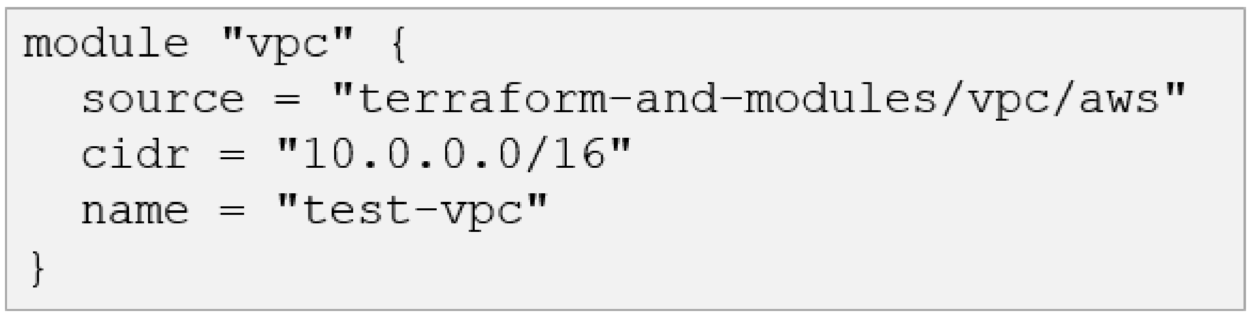
What must you do to successfully retrieve this value from your networking module?

A. Define the attribute vnet_id as a variable in the networking module
B. Change the referenced value to module.my_network.outputs.vnet_id
C. Define the attribute vnet_id as an output in the networking module
D. Change the referenced value to my_network.outputs.vnet_id

Answer: C

Explanation:
In a parent module, outputs of child modules are available in expressions as module.<MODULE NAME>.<OUTPUT NAME>. For example, if a child module named web_server declared an output named instance_ip_addr, you could access that value as module.web_server.instance_ip_addr.

NEW QUESTION 298
- (Exam Topic 4)
In the below configuration, how would you reference the module output vpc_id?



```
module "vpc" {
  source = "terraform-and-modules/vpc/aws"
  cidr = "10.0.0.0/16"
  name = "test-vpc"
}
```

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

A. Mastered
B. Not Mastered

Answer: A

Explanation:

https://cloudcasts.io/course/terraform/community-vpc-module

NEW QUESTION 302
- (Exam Topic 4)
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.

A. False
B. True

Answer: B

NEW QUESTION 307
- (Exam Topic 4)
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

Answer: D

Explanation:

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.

NEW QUESTION 311
- (Exam Topic 4)
Your firm employs a version control system (for example, git) and has requested that you commit all terraform code to it. During the commit, you must be cautious with sensitive information. Which of the following files should be left out of the commit?

A. main.tf
B. variables.tf
C. provisioner.tf
D. terraform.tfstate

Answer: D

NEW QUESTION 312
- (Exam Topic 4)
In the following code snippet, the block type is identified by which string?

A. "aws_instance"
B. resource
C. "db"
D. instance_type

Answer: B

NEW QUESTION 315
- (Exam Topic 4)
What does terraform refresh modify?

A. Your cloud infrastructure
B. Your state file
C. Your Terraform plan
D. Your Terraform configuration

Answer: B

Explanation:
The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match. Source: https://www.terraform.io/cli/commands/refresh

NEW QUESTION 317
- (Exam Topic 4)
Which of the following connection types are supported by the remote-exec provisioner? (select two)

A. WinRM
B. UDP
C. SMB
D. RDP
E. ssh

Answer: AE

Explanation:
The remote-exec provisioner invokes a script on a remote resource after it is created. The remote-exec provisioner supports both ssh and winrm type connections.
remote-exec connection types
* ssh on Linux
* winrm on Windows https://www.terraform.io/docs/provisioners/remote-exec.html

NEW QUESTION 319
- (Exam Topic 4)
What does terraform destroy do?

A. Destroy all infrastructure in the Terraform state file
B. Destroy all Terraform code files in the current directory while leaving the state file intact
C. Destroy all infrastructure in the configured Terraform provider
D. Destroy the Terraform state file while leaving infrastructure intact

Answer: A

Explanation:
The terraform destroy command terminates resources managed by your Terraform project. This command is the inverse of terraform apply in that it terminates all the resources specified in your Terraform state. It does not destroy resources running elsewhere that are not managed by the current Terraform project.
https://learn.hashicorp.com/tutorials/terraform/aws-destroy

NEW QUESTION 323
- (Exam Topic 4)
You have been working in a Cloud provider account that is shared with other team members. You previously used Terraform to create a load balancer that is listening on port 80. After some application changes, you updated the Terraform code to change the port to 443.
You run terraform plan and see that the execution plan shows the port changing from 80 to 443 like you intended, and step away to grab some coffee.
In the meantime, another team member manually changes the load balancer port to 443 through the Cloud provider console before you get back to your desk.
What will happen when you terraform apply upon returning to your desk?

A. Terraform will not make any changes to the Load Balancer and will update the state file to reflect any changes made.
B. Terraform will change the port back to 80 in your code
C. Terraform will change the load balancer port to 80, and) then change it back to 443
D. Terraform will fail with in error because the state file is no longer accurate

Answer: A

NEW QUESTION 327
......