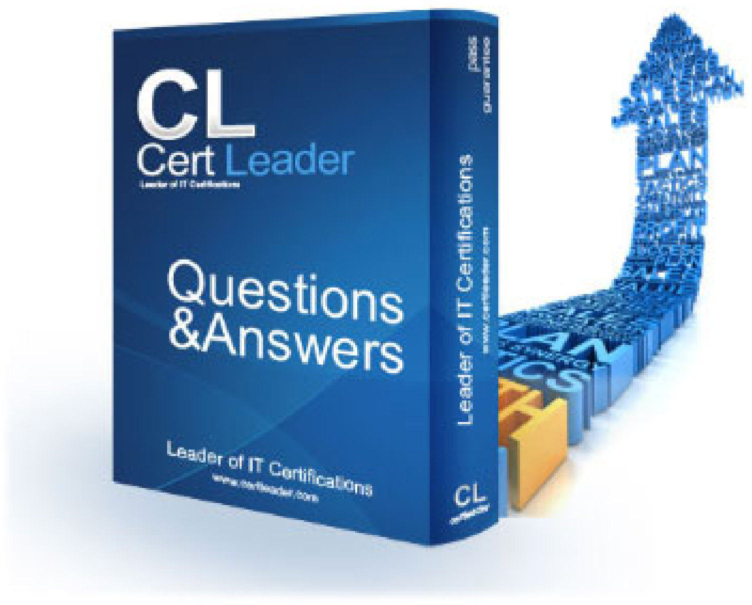
TA-002-P Dumps

HashiCorp Certified: Terraform Associate

https://www.certleader.com/TA-002-P-dumps.html



NEW QUESTION 1
- (Exam Topic 1)
What command should you run to display all workspaces for the current configuration?

A. terraform workspace
B. terraform workspace show
C. terraform workspace list
D. terraform show workspace

Answer: C

Explanation:

terraform workspace list
The command will list all existing workspaces.
Reference: https://www.terraform.io/docs/cli/commands/workspace/list.html

NEW QUESTION 2
- (Exam Topic 1)
Which of the following is available only in Terraform Enterprise or Cloud workspaces and not in Terraform CLI?

A. Secure variable storage
B. Support for multiple cloud providers
C. Dry runs with terraform plan
D. Using the workspace as a data source

Answer: A

Explanation:

Reference: https://www.terraform.io/docs/language/providers/configuration.html

NEW QUESTION 3
- (Exam Topic 1)
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 delete the VM using the cloud provider console, and run terraform apply again without changing any Terraform code?

A. Terraform will remove the VM from state file
B. Terraform will report an error
C. Terraform will not make any changes
D. Terraform will recreate the VM

Answer: D

NEW QUESTION 4
- (Exam Topic 1)
You have used Terraform to create an ephemeral development environment in the cloud 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 will be deleted by Terraform.
Which command should you use to show all of the resources that will be deleted? (Choose two.)

A. Run terraform plan -destroy.
B. This is not possibl
C. You can only show resources that will be created.
D. Run terraform state rm *.
E. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval.

Answer: AD

Explanation:

Reference: https://www.terraform.io/docs/cli/commands/state/rm.html

NEW QUESTION 5
- (Exam Topic 1)
When running the command terraform taint against a managed resource you want to force recreation upon, Terraform will immediately destroy and recreate the resource.

A. True
B. False

Answer: B

Explanation:
"The terraform taint command informs Terraform that a particular object has become degraded or damaged. Terraform represents this by marking the object as "tainted" in the Terraform state, and Terraform will propose to replace it in the next plan you create." FYI - This command is deprecated. For Terraform v0.15.2 and later, we recommend using the -replace option with terraform apply instead. For Terraform v0.15.2 and later, we recommend using the -replace option with

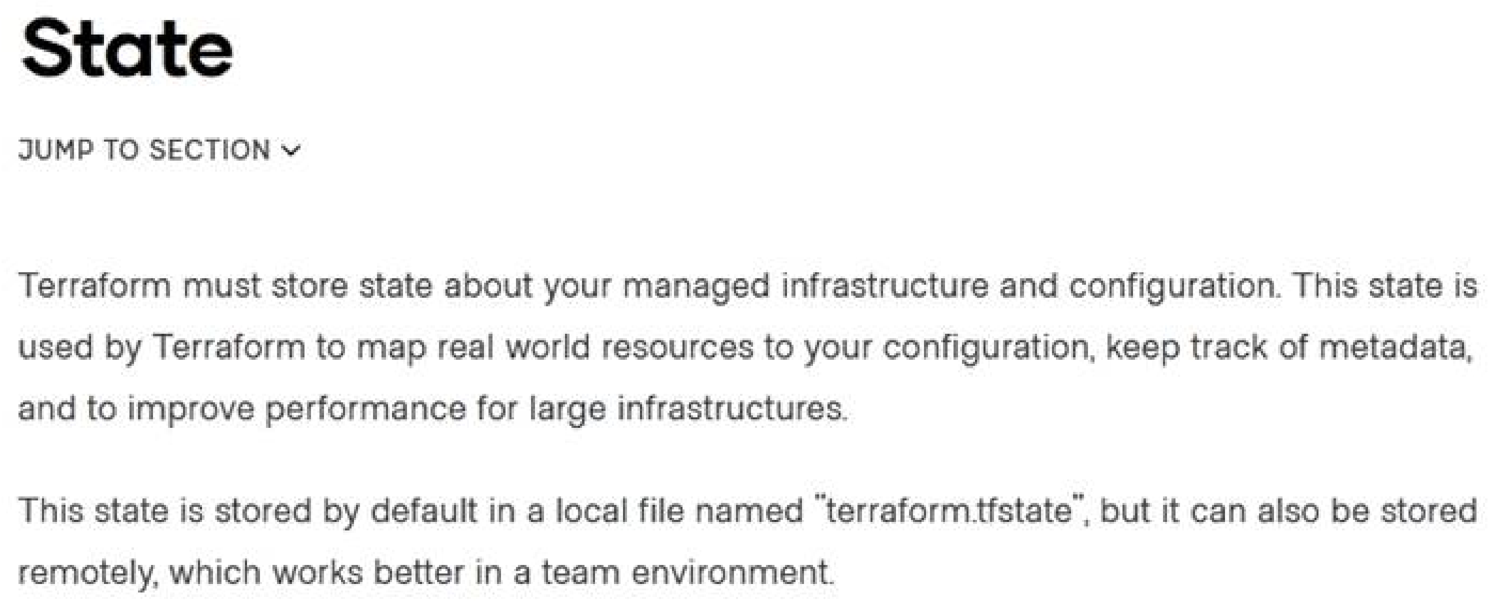
terraform apply to force Terraform to replace an object even though there are no configuration changes that would require it.
https://www.terraform.io/cli/commands/taint

NEW QUESTION 6
- (Exam Topic 1)
FILL BLANK
What is the name of the default file where Terraform stores the state?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
"This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment."
https://www.terraform.io/language/state


State

JUMP TO SECTION

Terraform must store state about your managed infrastructure and configuration. This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures.

This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

NEW QUESTION 7
- (Exam Topic 1)
How can terraform plan aid in the development process?

A. Validates your expectations against the execution plan without permanently modifying state
B. Initializes your working directory containing your Terraform configuration files
C. Formats your Terraform configuration files
D. Reconciles Terraform’s state against deployed resources and permanently modifies state using the current status of deployed resources

Answer: A

Explanation:
"The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. By default, when Terraform creates a plan it:
Reads the current state of any already-existing remote objects to make sure that the Terraform state is up-to-date.
Compares the current configuration to the prior state and noting any differences.
Proposes a set of change actions that should, if applied, make the remote objects match the configuration."
"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.
If Terraform detects that no changes are needed to resource instances or to root module output values, terraform plan will report that no actions need to be taken."
https://www.terraform.io/cli/commands/plan

NEW QUESTION 8
- (Exam Topic 1)
Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)

A. Destroy
B. Apply
C. Import
D. Init
E. Validate

Answer: BD

Explanation:
Reference: https://www.terraform.io/guides/core-workflow.html

NEW QUESTION 9
- (Exam Topic 1)
In contrast to Terraform Open Source, when working with Terraform Enterprise and Cloud Workspaces, conceptually you could think about them as completely separate working directories.

A. True
B. False

Answer: A

Explanation:
https://www.terraform.io/cloud-docs/workspaces
"When run locally, Terraform manages each collection of infrastructure with a persistent working directory, which contains a configuration, state data, and variables. Since Terraform CLI uses content from the directory it runs in, you can organize infrastructure resources into meaningful groups by keeping their configurations in separate directories.

NEW QUESTION 10
- (Exam Topic 1)
One remote backend configuration always maps to a single remote workspace.

A. True
B. False

Answer: B

Explanation:
The remote backend can work with either a single remote Terraform Cloud workspace, or with multiple similarly-named remote workspaces (like networking-dev and networking-prod). The workspaces block of the backend configuration determines which mode it uses: To use a single remote Terraform Cloud workspace, set workspaces.name to the remote workspace's full name (like networking-prod). To use multiple remote workspaces, set workspaces.prefix to a prefix used in all of the desired remote workspace names. For example, set prefix = "networking-" to use Terraform cloud workspaces with names like networking-dev and networking-prod. This is helpful when mapping multiple Terraform CLI workspaces used in a single Terraform configuration to multiple Terraform Cloud workspaces.

NEW QUESTION 10
- (Exam Topic 1)
Where does the Terraform local backend store its state?

A. In the /tmp directory
B. In the terraform.tfvars file
C. In the terraform.tfstate file
D. In the user's .terraformrc file

Answer: C

Explanation:
https://www.terraform.io/language/state
The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.
Reference: https://www.terraform.io/docs/language/settings/backends/local.html

NEW QUESTION 11
- (Exam Topic 1)
Which of the following is the correct way to pass the value in the variable num_servers into a module with the input servers?

A. servers = num_servers
B. servers = variable.num_servers
C. servers = var(num_servers)
D. servers = var.num_servers

Answer: D

Explanation:
"Within the module that declared a variable, its value can be accessed from within expressions as var.<NAME>, where <NAME> matches the label given in the declaration block:
Note: Input variables are created by a variable block, but you reference them as attributes on an object named var."
https://www.terraform.io/language/values/variables#using-input-variable-values

NEW QUESTION 14
- (Exam Topic 1)
You have never used Terraform before and would like to test it out using a shared team account for a cloud provider. The shared team account already contains 15 virtual machines (VM). You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully. What should you do to delete the newly-created VM with Terraform?

A. The Terraform state file contains all 16 VMs in the team accoun
B. Execute terraform destroy and select the newly-created VM.
C. The Terraform state file only contains the one new V
D. Execute terraform destroy.
E. Delete the Terraform state file and execute Terraform apply.
F. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.

Answer: B

Explanation:
You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully. read the question carefully "Terraform configuration containing one VM, perform terraform apply" so only one VM is in state file.

NEW QUESTION 18
- (Exam Topic 1)
You would like to reuse the same Terraform configuration for your development and production environments with a different state file for each.
Which command would you use?

A. terraform import
B. terraform workspace
C. terraform state
D. terraform init

Answer: B

Explanation:
https://www.terraform.io/language/state/workspaces#when-to-use-multiple-workspaces

NEW QUESTION 23
- (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 27
- (Exam Topic 1)
HashiCorp Configuration Language (HCL) supports user-defined functions.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/language/functions
The Terraform language does not support user-defined functions, and so only the functions built into the language are available for use

NEW QUESTION 28
- (Exam Topic 1)
Which of the following is not true of Terraform providers?

A. Providers can be written by individuals
B. Providers can be maintained by a community of users
C. Some providers are maintained by HashiCorp
D. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers
E. None of the above

Answer: E

Explanation:
https://registry.terraform.io/providers/hashicorp/google/latest - This provider is collaboratively maintained by the Google Terraform Team at Google and the Terraform team at HashiCorp
https://www.terraform.io/language/providers

NEW QUESTION 32
- (Exam Topic 1)
How is the Terraform remote backend different than other state backends such as S3, Consul, etc.?

A. It can execute Terraform runs on dedicated infrastructure on premises or in Terraform Cloud
B. It doesn't show the output of a terraform apply locally
C. It is only available to paying customers
D. All of the above

Answer: A

Explanation:
Backends define where Terraform's state snapshots are stored. A given Terraform configuration can either specify a backend, integrate with Terraform Cloud, or do neither and default to storing state locally.
If you and your team are using Terraform to manage meaningful infrastructure, we recommend using the remote backend with Terraform Cloud or Terraform Enterprise.
Reference: https://www.terraform.io/docs/language/settings/backends/index.html

NEW QUESTION 35

- (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 39
- (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 40
- (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 42
- (Exam Topic 2)
Which of the below are paid features of Terraform Cloud?

A. Full API Coverage
B. Secure variable Storage
C. Roles/ Team management
D. Cost Estimation
E. Private Module Registry
F. Sentinel policies

Answer: CDF

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

NEW QUESTION 44
- (Exam Topic 2)
The Terraform language does not support user-defined functions, and so only the functions built in to the language are available for use.

A. False
B. True

Answer: B

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

NEW QUESTION 45

- (Exam Topic 2)
You want to use different AMI images for different regions and for the purpose you have defined following code block.
* 1.variable "images"
* 2.{
* 3. type = "map"
* 4.
* 5. default = {
* 6. us-east-1 = "image-1234"
* 7. us-west-2 = "image-4567"
* 8. us-west-1 = "image-4589"
* 9. }
* 10.}
What of the following approaches needs to be followed in order to select image-4589?

A. var.images["us-west-1"]
B. var.images[3]
C. var.images[2]
D. lookup(var.images["us-west-1"]

Answer: A

NEW QUESTION 50
- (Exam Topic 2)
Matt wants to import a manually created EC2 instance into terraform so that he can manage the EC2 instance through terraform going forward. He has written the configuration file of the EC2 instance before importing it to Terraform. Following is the code:
resource "aws_instance" "matt_ec2" { ami = "ami-bg2640de" instance_type = "t2.micro" vpc_security_group_ids = ["sg-6ae7d613", "sg-53370035"] key_name = "mysecret" subnet_id =
"subnet-9e3cfbc5" }
The instance id of that EC2 instance is i-0260835eb7e9bd40 How he can import data of EC2 to state file?

A. terraform import aws_instance.id = i-0260835eb7e9bd40
B. terraform import i-0260835eb7e9bd40
C. terraform import aws_instance.i-0260835eb7e9bd40
D. terraform import aws_instance.matt_ec2 i-0260835eb7e9bd40

Answer: D

Explanation:
https://www.terraform.io/docs/import/usage.html

NEW QUESTION 55
- (Exam Topic 2)
Terraform has detailed logs which can be enabled by setting the _____ environmental variable.

A. TF_TRACE
B. TF_DEBUG
C. TF_LOG
D. TF_INFO

Answer: C

Explanation:
Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr.
You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name. https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION 56
- (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 59
- (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 60
- (Exam Topic 2)
terraform state subcommands such as list are read-only commands, do read-only commands create state backup files?

A. Yes
B. No

Answer: B

Explanation:
Subcommands that are read-only (such as list) do not write any backup files since they aren't modifying the state.
All terraform state subcommands that modify the state write backup files. The path of these backup file can be controlled with -backup.
https://www.terraform.io/docs/commands/state/index.html#backups

NEW QUESTION 62
- (Exam Topic 2)
ABC Enterprise has recently tied up with multiple small organizations for exchanging database information. Due to this, the firewall rules are increasing and are more than 100 rules. This is leading firewall configuration file that is difficult to manage. What is the way this type of configuration can be managed easily?

A. Terraform Backends
B. Terraform Functions
C. Dynamic Blocks
D. Terraform Expression

Answer: C

NEW QUESTION 67
- (Exam Topic 2)
You have created 2 workspaces PROD and RQA. You have switched to RQA and provisioned RQA infrastructure from this workspace. Where is your state file stored?

A. terraform.tfstate.d
B. terraform.d
C. terraform.tfstate.RQA
D. terraform.tfstate

Answer: A

NEW QUESTION 70
- (Exam Topic 2)
lookup retrieves the value of a single element from which of the below data type?

A. map
B. set
C. string
D. list

Answer: A

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

NEW QUESTION 72
- (Exam Topic 2)
Which Terraform command will force a marked resource to be destroyed and recreated on the next apply?

A. terraform fmt
B. terraform destroy
C. terraform taint
D. terraform refresh

Answer: C

Explanation:
The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.
Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.
https://www.terraform.io/docs/commands/taint.html

NEW QUESTION 74
- (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 76
- (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 78
- (Exam Topic 2)
You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?

A. It depends on which command issue related to.
B. Python
C. Go
D. Java

Answer: C

Explanation:
Basic programming knowledge. Terraform and Terraform Plugins are written in the Go programming language, but even if you've never written a line of Go before, you're still welcome to take a dive into the code and submit patches. The community is happy to assist with code reviews and offer guidance specific to Go.

NEW QUESTION 82
- (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 85
- (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 86
- (Exam Topic 2)
Which of the below configuration file formats are supported by Terraform? (Select TWO)

A. Node
B. JSON
C. Go
D. YAML
E. HCL

Answer: BE

Explanation:
Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations. https://www.terraform.io/docs/configuration/

NEW QUESTION 88
- (Exam Topic 2)
Which of the following command can be used to view the specified version constraints for all providers used in the current configuration.

A. terraform providers
B. terraform state show
C. terraform provider
D. terraform plan

Answer: A

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

NEW QUESTION 89
- (Exam Topic 2)
What is the command you can use to set an environment variable named "var1"of type String?

A. export TF_VAR_VAR1
B. set TF_VAR_var1
C. variable "var1" { type = "string"}
D. export TF_VAR_var1

Answer: D

Explanation:
The environment variable must be in the format TF_VAR_name, so for the QUESTION NO: TF_VAR_var1 is the correct choice.
https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION 91
- (Exam Topic 2)
Terraform must track metadata such as resource dependencies. Where is this data stored?

A. workspace
B. backend
C. state file
D. metadata store

Answer: C

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

NEW QUESTION 94
- (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 99
- (Exam Topic 3)
Which of the following is the right substitute for static values that can make Terraform configuration file more dynamic and reusable?

A. Output value
B. Input parameters
C. Functions
D. Modules

Answer: B

Explanation:
Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.

NEW QUESTION 101
- (Exam Topic 3)
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. What command will do this?

A. terraform taint
B. terraform apply
C. terraform graph
D. terraform refresh

Answer: A

Explanation:
The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.
Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.
This example will taint a single resource:
$ terraform taint aws_security_group.allow_all
The resource aws_security_group.allow_all in the module root has been marked as tainted. https://www.terraform.io/docs/commands/taint.html

NEW QUESTION 102
- (Exam Topic 3)
Which of the below features of Terraform can be used for managing small differences between different environments which can act more like completely separate working directories.

A. Repositories
B. Workspaces
C. Environment Variables
D. Backends

Answer: B

Explanation:
workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. They are convenient in a number of situations, but cannot solve all problems.
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. For example, a developer working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.
Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure. When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that
changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.
https://www.terraform.io/docs/state/workspaces.html https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces

NEW QUESTION 103
- (Exam Topic 3)
You have multiple developers working on a terraform project (using terraform OSS), and have saved the terraform state in a remote S3 bucket . However ,team is intermittently experiencing inconsistencies in the provisioned infrastructure / failure in the code . You have traced this problem to simultaneous/concurrent runs of terraform apply command for 2/more developers . What can you do to fix this problem?

A. Use terraform workspaces feature, this will fix this problem by default , as every developer will have their own state file , and terraform will merge them on server side on its own.
B. Structure your team in such a way that only one individual will run terraform apply , everyone will just make changes and share with hi
C. Then there will be no chance of any inconsistencies.
D. Stop using remote state , and store the developer tfstate in their own machine . Once a day , all developers should sit together and merge the state files manually , to avoid any inconsistencies.
E. Enable terraform state locking for the S3 backend using DynamoDB tabl
F. This prevents others from acquiring the lock and potentially corrupting your state.

Answer: D

Explanation:
S3 backend support state locking using DynamoDB. https://www.terraform.io/docs/state/locking.html

NEW QUESTION 107
- (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 112
- (Exam Topic 3)
You cannot publish your own modules on the Terraform Registry.

A. False
B. True

Answer: A

Explanation:
Anyone can publish and share modules on the Terraform Registry. https://www.terraform.io/docs/registry/modules/publish.html

NEW QUESTION 115
- (Exam Topic 3)
You want terraform plan and terraform apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?

A. Local Backends.
B. Terraform Backends.
C. This can be done using any of the local or remote backends.
D. Remote Backends.

Answer: D

Explanation:
When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.
Terraform Cloud can also be used with local operations, in which case only state is stored in the Terraform Cloud backend.
https://www.terraform.io/docs/backends/types/remote.html

NEW QUESTION 119
- (Exam Topic 3)
Hanah is writing a terraform configuration with nested modules, there are multiple places where she has to use the same conditional expression but she wants to avoid repeating the same values or expressions multiple times in the configuration,. What is a better approach to dealing with this?

A. Expressions
B. Local Values
C. Variables
D. Functions

Answer: B

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

NEW QUESTION 121
- (Exam Topic 3)
By default, provisioners that fail will also cause the Terraform apply itself to error. How can you change this default behavior within a provisioner?

A. provisioner "local-exec" { on_failure = "next" }
B. provisioner "local-exec" { when = "failure" terraform apply }
C. provisioner "local-exec" { on_failure = "continue" }
D. provisioner "local-exec" { on_failure = continue }

Answer: C

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

NEW QUESTION 123
- (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 126
- (Exam Topic 3)
You have created a terraform script that uses a lot of new constructs that have been introduced in terraform v0.12. However, many developers who are cloning the script from your git repo, are using v0.11, and getting errors. What can be done from your end to solve this problem?

A. Force developer to use v0.12 by using terraform setting ‘required_version’ and set it to >=0.12.
B. Refactor the code to support both v0.11, and v0.12. It might be a difficult process, but there is no other way.
C. Add a condition in front of each such specific construct, to check whether the running terraform version id v0.11 or v0.12, and ,work accordingly.
D. Add comments in your code to tell developers to use v0.12 . If they use v0.11 , that should be their problem , which they need to figure out.

Answer: A

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

NEW QUESTION 130
- (Exam Topic 3)
Why is it a good idea to declare the required version of a provider in a Terraform configuration file?
* 1. terraform
* 2. {
* 3. required_providers
* 4. {
* 5. aws = "~> 1.0"
* 6. }
* 7. }

A. To remove older versions of the provider.
B. To ensure that the provider version matches the version of Terraform you are using.
C. Providers are released on a separate schedule from Terraform itself; therefore a newer version could introduce breaking changes.
D. To match the version number of your application being deployed via Terraform.

Answer: C

NEW QUESTION 134
- (Exam Topic 3)
What does terraform refresh command do?

A. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
B. terraform refresh command basically updates the configuration file with the current state of the actual infrastructure
C. terraform refresh is use to change/modify the infrastructure based on the existing state file, at that moment.
D. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
E. terraform refresh syncs the state file with the real world infrastructure.

Answer: E

NEW QUESTION 139
- (Exam Topic 4)
How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? (Choose two.)

A. A full audit trail of the request and fulfillment process is generated
B. A request must be submitted for infrastructure changes
C. As additional resources are required, more tickets are submitted
D. A catalog of approved resources can be accessed from drop down lists in a request form

Answer: BC

NEW QUESTION 141
- (Exam Topic 4)
Select the most accurate statement to describe the Terraform language from the following list.

A. Terraform is an immutable, declarative, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally JSON.
B. Terraform is a mutable, declarative, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
C. Terraform is an immutable, procedural, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
D. Terraform is a mutable, procedural, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally YAML.

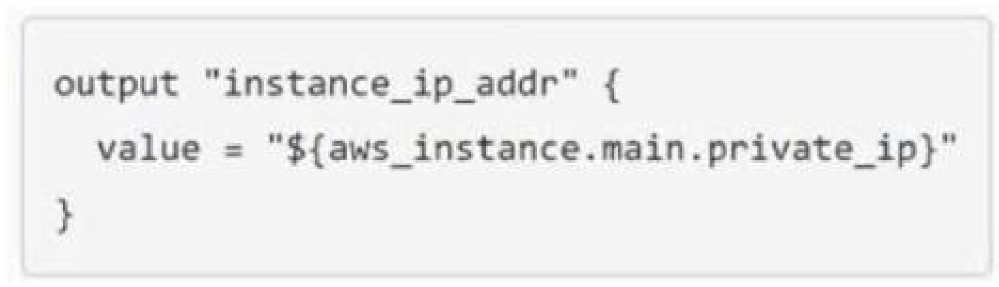
Answer: A

Explanation:

Terraform is not a configuration management tool - https://www.terraform.io/intro/vs/chefpuppet.html Terraform is a declarative language - https://www.terraform.io/docs/configuration/index.html Terraform supports a syntax that is JSON compatible https://www.terraform.io/docs/configuration/syntax-json.html
Terraform is primarily designed on immutable infrastructure principles - https://www.hashicorp.com/resources/what-is-mutable-vs-immutable-infrastructure

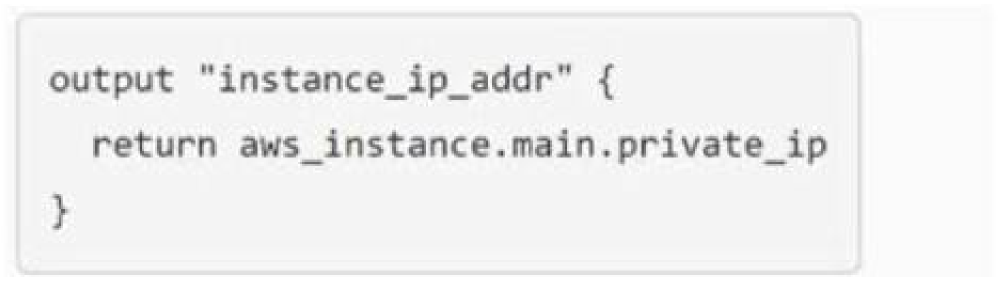
NEW QUESTION 146
- (Exam Topic 4)
You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name the instance resource "main".
Which of these is the correct way to define the output value using HCL2?

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 149
- (Exam Topic 4)
True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.

A. False
B. True

Answer: A

Explanation:

Currently, Terraform has no mechanism to redact or protect secrets that are returned via data sources, so secrets read via this provider will be persisted into the Terraform state, into any plan files, and in some cases in the console output produced while planning and applying. These artifacts must, therefore, all be protected accordingly.

NEW QUESTION 154
- (Exam Topic 4)
Which of the following value will be accepted for my_var?
* 1. variable "my_var"
* 2. {
* 3. type = string
* 4. }

A. 15
B. "15"
C. Both A and B
D. None of the above

Answer: C

Explanation:
The Terraform language will automatically convert number and bool values to string values when needed, and vice-versa as long as the string contains a valid representation of a number or boolean value. Example
* true converts to "true", and vice-versa
* false converts to "false", and vice-versa
* 15 converts to "15", and vice-versa
Where possible, Terraform automatically converts values from one type to another in order to produce the expected type. If this isn't possible, Terraform will produce a type mismatch error and you must update the
configuration with a more suitable expression. https://www.terraform.io/docs/configuration/expressions.html#type-conversion

NEW QUESTION 159

- (Exam Topic 4)
Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)

A. Explicit use of a provider block in configuration, optionally including a version constraint.
B. Use of any resource belonging to a particular provider in a resource or data block in configuration.
C. Existence of any resource instance belonging to a particular provider in the current state.
D. Existence of any provider plugins found locally in the working directory.

Answer: ABC

Explanation:
The existence of a provider plugin found locally in the working directory does not itself create a provider dependency. The plugin can exist without any reference to it in the terraform configuration. https://www.terraform.io/docs/commands/providers.html

NEW QUESTION 164
- (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 165
- (Exam Topic 4)
When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)
* 1. terraform
* 2. {
* 3. required_providers
* 4. {
* 5. aws = "~> 1.2.0"
* 6. }
* 7. }

A. 1.3.1
B. 1.2.3
C. 1.2.9
D. 1.3.0

Answer: BC

Explanation:
As your Terraform usage becomes more advanced, there are some cases where you may need to modify the Terraform state. Rather than modify the state directly, the terraform state commands can be used in many cases instead. This command is a nested subcommand, meaning that it has further subcommands.
https://www.terraform.io/docs/commands/state/index.html

NEW QUESTION 166
- (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 168
- (Exam Topic 4)
Which parameters does terraform import require? Choose two correct answers.

A. Provider
B. Path
C. Resource address
D. Resource ID

Answer: CD

Explanation:
https://www.terraform.io/cli/commands/import#usage

NEW QUESTION 170
- (Exam Topic 4)
In the example below, where is the value of the DNS record's IP address originating from?
* 1. resource "aws_route53_record" "www"

* 2. {
* 3. zone_id = aws_route53_zone.primary.zone_id
* 4. name = "www.example.com"
* 5. type = "A"
* 6. ttl = "300"
* 7. records = [module.web_server.instance_ip_address] 8. }

A. The regular expression named module.web_server
B. The output of a module named web_server
C. By querying the AWS EC2 API to retrieve the IP address
D. Value of the web_server parameter from the variables.tf file

Answer: B

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_address, you could access that value as module.web_server.instance_ip_address.

NEW QUESTION 172
- (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 176
- (Exam Topic 4)
From the code below, identify the implicit dependency:

A. The EIP with an id of ami-2757f631
B. The AMI used for the EC2 instance
C. The EC2 instance labeled web_server
D. The S3 bucket labeled company_data

Answer: C

NEW QUESTION 179
- (Exam Topic 4)
Which statements best describes what the local variable assignment is doing in the following code snippet:

A. Create a distinct list of route table name objects
B. Create a map of route table names to subnet names
C. Create a map of route table names from a list of subnet names
D. Create a list of route table names eliminating duplicates

Answer: D

NEW QUESTION 180
- (Exam Topic 4)
Multiple provider instances blocks for AWS can be part of a single configuration file?

A. False
B. True

Answer: B

Explanation:
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:
# The default provider configuration provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region provider "aws" {
alias = "west" region = "us-west-2"
}
The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances

NEW QUESTION 181
- (Exam Topic 4)
Which of the following locations can Terraform use as a private source for modules? (Choose two.)

A. Internally hosted SCM (Source Control Manager) platform
B. Public Terraform Module Registry
C. Private repository on GitHub
D. Public repository on GitHub

Answer: AC

NEW QUESTION 186
- (Exam Topic 4)
What resource dependency information is stored in Terraform's state?

A. Only implicit dependencies are stored in state.
B. Both implicit and explicit dependencies are stored in state.
C. Only explicit dependencies are stored in state.
D. No dependency information is stored in state.

Answer: B

Explanation:
Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state. https://www.terraform.io/docs/state/purpose.html#metadata

NEW QUESTION 191
- (Exam Topic 4)
Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements. (select three)

A. The module must be PCI/HIPPA compliant.
B. Module repositories must use this three-part name format, terraform-- .
C. The registry uses tags to identify module versions.
D. Release tag names must be for the format x.y.z, and can optionally be prefixed with a v .
E. The module must be on GitHub and must be a public repo.

Answer: CDE

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

NEW QUESTION 196
- (Exam Topic 4)
A Terraform backend determines how Terraform loads state and stores updates when you execute _____.

A. apply
B. taint
C. destroy
D. All of the above
E. None of the above

Answer: D

NEW QUESTION 197
- (Exam Topic 4)
terraform destroy is the only way to remove infrastructure.

A. True
B. False

Answer: B

NEW QUESTION 198
- (Exam Topic 4)
All modules published on the official Terraform Module Registry have been verified by HashiCorp.

A. True
B. False

Answer: B

Explanation:
https://registry.terraform.io/
Only modules considered "Verified Modules" are reviewed by Hashicorp, otherwise anyone can publish modules on the Terraform Registry.
Reference: https://www.terraform.io/registry/modules/verified https://www.terraform.io/registry/modules/publish

NEW QUESTION 203
- (Exam Topic 4)
Your team uses terraform OSS . You have created a number of resuable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option/way to do that?

A. Terraform modules cannot be shared in OSS version . Each developer needs to maintain their own modules and leverage them in the main tf file.
B. Upload your modules with proper versioning in the terraform public module registry . Terraform OSS is directly integrated with the public module registry , and can reference the modules from the code in the main tf file.
C. Terraform module sharing is only available in Enterprise version via terraform private module registry , so no way to enable it in OSS version.
D. Store your modules in a NAS/ shared file server , and ask your team members to directly reference thecode from ther
E. This is the only viable option in terraform OSS ,which is better than individually maintaining module versions for every developer.

Answer: B

Explanation:
Software development encourages code reuse through reusable artifacts, such as libraries, packages and modules. Most programming languages enable developers to package and publish these reusable components and make them available on a registry or feed. For example, Python has Python Package Index and PowerShell has PowerShell Gallery.
For Terraform users, the Terraform Registry enables the distribution of Terraform modules, which are reusable configurations. The Terraform Registry acts as a centralized repository for module sharing, making modules easier to discover and reuse.
The Registry is available in two variants:
* Public Registry houses official Terraform providers -- which are services that interact with an API to expose and manage a specific resource -- and community-contributed modules.
* Private Registry is available as part of the Terraform Cloud, and can host modules internally within an organization.
https://www.terraform.io/docs/registry/index.html

NEW QUESTION 205
- (Exam Topic 4)
Do terraform workspaces help in adding/allowing multiple state files for a single configuration?

A. True
B. False

Answer: A

NEW QUESTION 209
- (Exam Topic 4)
You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed?

A. Terraform can still determine the correct order for destruction from the state even when you delete one or more items from the configuration.
B. Those would be destroyed in the order in which they were written in the configuration file previously before you have deleted them from configuration file.
C. The resource will be destructed in random order as you have already deleted them from configuration.
D. You can not destroy resources by deleting them from configuration file and running plan and apply.

Answer: A

Explanation:
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.

NEW QUESTION 213
- (Exam Topic 4)
Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

A. Unlock
B. force-unlock
C. Removing the lock on a state file is not possible
D. state-unlock

Answer: B

Explanation:

https://www.terraform.io/docs/commands/force-unlock.html

NEW QUESTION 218
- (Exam Topic 4)
Which of the following statements about local modules is incorrect:

A. Local modules are not cached by terraform init command
B. Local modules are sourced from a directory on disk
C. Local modules support versions
D. All of the above (all statements above are incorrect
E. None of the above (all statements above are correct)

Answer: C

Explanation:
Version constraints are supported only for modules installed from a module registry, such as the public Terraform Registry or Terraform Cloud's private module registry. Other module sources can provide their own versioning mechanisms within the source string itself, or might not support versions at all. In particular, modules sourced from local file paths do not support version; since they're loaded from the same source repository, they always share the same version as their caller.
https://www.terraform.io/language/modules/syntax

NEW QUESTION 221
- (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 225
- (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 226
- (Exam Topic 4)
Which of the following is a meta-argument defined in the configuration files of Terraform?

A. tfvar
B. depends_on
C. instance aws
D. varl

Answer: B

NEW QUESTION 230
- (Exam Topic 4)
In order to reduce the time it takes to provision resources, Terraform uses parallelism. By default, how many resources will Terraform provision concurrently?

A. 5
B. 50
C. 10
D. 20

Answer: C

NEW QUESTION 234
- (Exam Topic 4)
Which of the following is considered a Terraform plugin?

A. Terraform language

B. Terraform tooling
C. Terraform logic
D. Terraform provider

Answer: D

Explanation:
Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins, even the core types such as AWS and Heroku. Users of Terraform are able to write new plugins in order to support new functionality in Terraform.
https://www.terraform.io/docs/plugins/basics.html

NEW QUESTION 238
- (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 240
- (Exam Topic 4)
Using multi-cloud and provider-agnostic tools provides which of the following benefits?

A. Operations teams only need to learn and manage a single tool to manage infrastructure, regardless of where the infrastructure is deployed.
B. Increased risk due to all infrastructure relying on a single tool for management.
C. Can be used across major cloud providers and VM hypervisors.
D. Slower provisioning speed allows the operations team to catch mistakes before they are applied.

Answer: AC

Explanation:
Using a tool like Terraform can be advantageous for organizations deploying workloads across multiple public and private cloud environments. Operations teams only need to learn a single tool, single language, and can use the same tooling to enable a DevOps-like experience and workflows.

NEW QUESTION 243
- (Exam Topic 4)
What Terraform command can be used to inspect the current state file?

A. terraform inspect
B. terraform read
C. terraform show
D. terraform state

Answer: C

NEW QUESTION 248
- (Exam Topic 4)
Which are forbidden actions when the Terraform state file is locked? (Choose three.)

A. terraform destroy
B. terraform fmt
C. terraform state list
D. terraform apply
E. terraform plan
F. terraform validate

Answer: ADE

NEW QUESTION 249
- (Exam Topic 4)
While attempting to deploy resources into your cloud provider using Terraform. you begin to see some odd behavior and experience sluggish responses. In order to troubleshoot you decide to turn on Terraform debugging. Which environment variables must be configured to make Terraform's logging more verbose?

A. TF_10G_PATM
B. TF_LOG
C. TF_10G_LEVEL
D. TF.LOG.FUE

Answer:

B

Explanation:
https://www.terraform.io/internals/debugging

NEW QUESTION 254
- (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 257
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

Answer: A

NEW QUESTION 259
- (Exam Topic 4)
Given the Terraform configuration below, in which order will the resources be created?
* 1. resource "aws_instance" "web_server"
* 2. {
* 3. ami = "ami-b374d5a5"
* 4. instance_type = "t2.micro"
* 5. }
* 6. resource "aws_eip" "web_server_ip"
* 7. {
* 8. vpc = true instance = aws_instance.web_server.id
* 9. }

A. aws_eip will be created first aws_instance will be created second
B. aws_eip will be created first aws_instance will be created second
C. Resources will be created simultaneously
D. aws_instance will be created first aws_eip will be created second

Answer: D

Explanation:
Implicit and Explicit Dependencies
By studying the resource attributes used in interpolation expressions, Terraform can automatically infer when one resource depends on another. In the example above, the reference to aws_instance.web_server.id creates an implicit dependency on the aws_instance named web_server.
Terraform uses this dependency information to determine the correct order in which to create the different resources.
# Example of Implicit Dependency resource "aws_instance" "web_server" { ami = "ami-b374d5a5"
instance_type = "t2.micro"
}
resource "aws_eip" "web_server_ip" { vpc = true
instance = aws_instance.web_server.id
}
In the example above, Terraform knows that the aws_instance must be created before the aws_eip. Implicit dependencies via interpolation expressions are the primary way to inform Terraform about these
relationships, and should be used whenever possible.
Sometimes there are dependencies between resources that are not visible to Terraform. The depends_on argument is accepted by any resource and accepts a list of resources to create explicit dependencies for.
For example, perhaps an application we will run on our EC2 instance expects to use a specific Amazon S3 bucket, but that dependency is configured inside the application code and thus not visible to Terraform. In that case, we can use depends_on to explicitly declare the dependency:
# Example of Explicit Dependency
# New resource for the S3 bucket our application will use. resource "aws_s3_bucket" "example" {
bucket = "terraform-getting-started-guide" acl = "private"
}
# Change the aws_instance we declared earlier to now include "depends_on" resource "aws_instance" "example" {
ami = "ami-2757f631" instance_type = "t2.micro"
# Tells Terraform that this EC2 instance must be created only after the
# S3 bucket has been created. depends_on = [aws_s3_bucket.example]
}
https://learn.hashicorp.com/terraform/getting-started/dependencies.html

NEW QUESTION 260

- (Exam Topic 4)
Which of the following is not an advantage of using infrastructure as code operations?

A. Self-service infrastructure deployment
B. Troubleshoot via a Linux diff command
C. Public cloud console configuration workflows
D. Modify a count parameter to scale resources
E. API driven workflows

Answer: B

Explanation:
terraform is used to deploy the infrastructure, not to troubleshoot it

NEW QUESTION 264
- (Exam Topic 4)
Terraform Cloud is available only as a paid offering from HashiCorp.

A. True
B. False

Answer: B

Explanation:
Many of Terraform Cloud features are free for small teams, including remote state storage, remote runs, and VCS connections.
"Terraform Cloud is a commercial SaaS product developed by HashiCorp. Many of its features are free for small teams, including remote state storage, remote runs, and VCS connections. We also offer paid plans for larger teams that include additional collaboration and governance features."

NEW QUESTION 269
- (Exam Topic 4)
What command can you run to generate DOT (Document Template) formatted data to visualize Terraform dependencies?

A. terraform refresh
B. terraform show
C. terraform graph
D. terraform output

Answer: C

Explanation:
The terraform graph command is used to generate a visual representation of either a configuration or execution plan. The output is in the DOT format, which can be used by GraphViz to generate charts.

NEW QUESTION 272
- (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 274
- (Exam Topic 4)
You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration?
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:
Once you have authenticated to Terraform Cloud, you're ready to migrate your local state file to Terraform Cloud. To begin the migration, reinitialize. This causes Terraform to recognize your cloud block configuration.

NEW QUESTION 276
- (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 278
- (Exam Topic 4)
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.

A. False
B. True

Answer: B

NEW QUESTION 281
- (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 286
- (Exam Topic 4)
You have just developed a new Terraform configuration for two virtual machines with a cloud provider. You would like to create the infrastructure for the first time. Which Terraform command should you run first?

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

Answer: D

NEW QUESTION 288
- (Exam Topic 4)
What Terraform feature is shown in the example below?

A. conditional expression
B. local values
C. dynamic block
D. data source

Answer: C

NEW QUESTION 292
- (Exam Topic 4)
Which of the following is not a way to trigger terraform destroy ?

A. Passing ---destroy at the end of apian request
B. Running terraform destroy from the correct directory and then typing "yes" when prompted in the CLI
C. Using the destroy command with auto approve
D. Delete the state file and run terraform apply

Answer: A

NEW QUESTION 296
......

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your TA-002-P Exam with Our Prep Materials Via below:

https://www.certleader.com/TA-002-P-dumps.html