Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

https://www.2passeasy.com/dumps/TA-002-P/

NEW QUESTION 1
- (Exam Topic 1)
A fellow developer on your team is asking for some help in refactoring their Terraform code. As part of their application's architecture, they are going to tear down an existing deployment managed by Terraform and deploy new. However, there is a server resource named aws_instance.ubuntu[1] they would like to keep to perform some additional analysis.
What command should be used to tell Terraform to no longer manage the resource?

A. terraform apply rm aws_instance.ubuntu[1]
B. terraform state rm aws_instance.ubuntu[1]
C. terraform plan rm aws_instance.ubuntu[1]
D. terraform delete aws_instance.ubuntu[1]

Answer: B

Explanation:
"You can use terraform state rm in the less common situation where you wish to remove a binding to an existing remote object without first destroying it, which will effectively make Terraform "forget" the object while it continues to exist in the remote system." https://www.terraform.io/cli/commands/state/rm

NEW QUESTION 2
- (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 3
- (Exam Topic 1)
Which of the following is not a valid string function in Terraform?

A. split
B. join
C. slice
D. chomp

Answer: C

Explanation:

https://www.terraform.io/language/functions

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)
Terraform can only manage resource dependencies if you set them explicitly with the depends_on argument.

A. True
B. False

Answer: A

Explanation:
"Use the depends_on meta-argument to handle hidden resource or module dependencies that Terraform cannot automatically infer. You only need to explicitly specify a dependency when a resource or module relies on another resource's behavior but does not access any of that resource's data in its arguments."
https://www.terraform.io/language/meta-arguments/depends_on

NEW QUESTION 6

- (Exam Topic 1)
Terraform providers are always installed from the Internet.

A. True
B. False

Answer: B

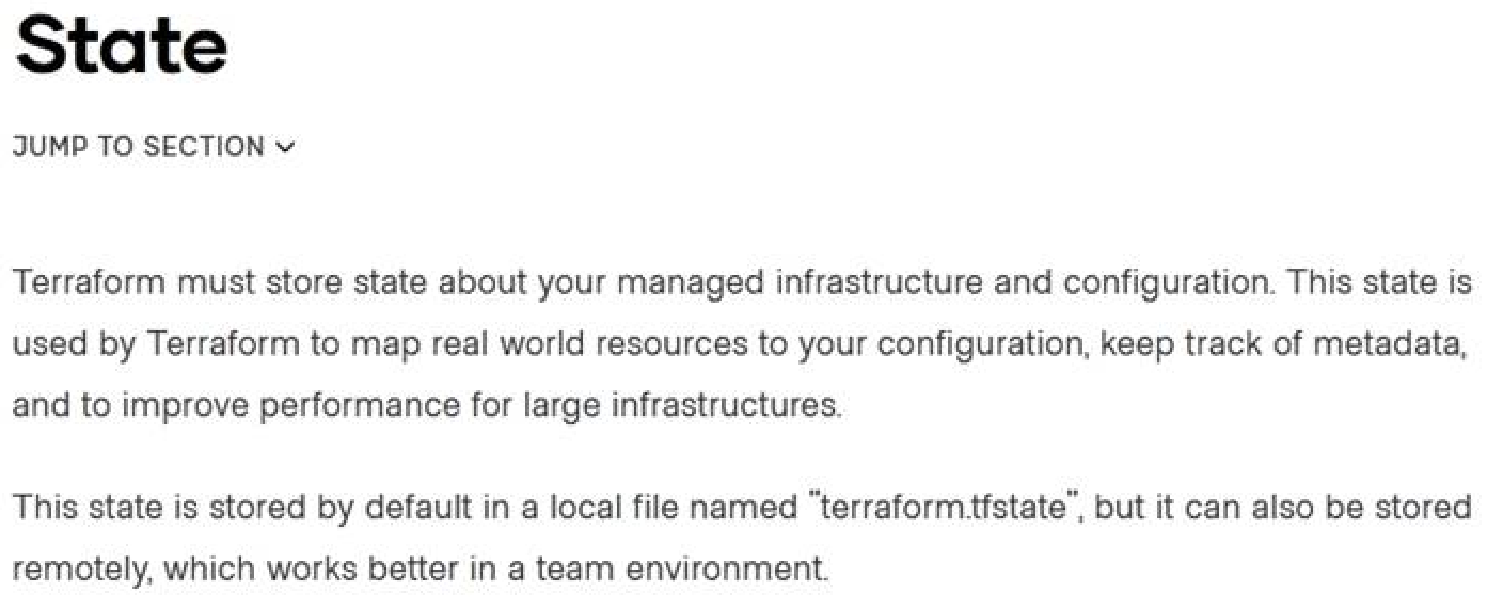
Explanation:
Terraform configurations must declare which providers they require, so that Terraform can install and use them.
Reference: https://www.terraform.io/docs/language/providers/configuration.html

NEW QUESTION 7
- (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 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)
A terraform apply can not ______ infrastructure.

A. change
B. destroy
C. provision
D. import

Answer: D

Explanation:
https://www.educative.io/answers/what-is-the-command-to-destroy-infrastructure-in-terraform

NEW QUESTION 10
- (Exam Topic 1)
Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/downloads

NEW QUESTION 10
- (Exam Topic 1)
You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (files). You need to enable debug messages to find this out.
Which of the following would achieve this?

A. Set the environment variable TF_LOG=TRACE
B. Set verbose logging for each provider in your Terraform configuration
C. Set the environment variable TF_VAR_log=TRACE
D. Set the environment variable TF_LOG_PATH

Answer: A

Explanation:
Although this will only output to stderr and if you need to review log file you will need to include TF_LOG_PATH=pathtofile
https://www.terraform.io/internals/debugging

NEW QUESTION 12
- (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 14
- (Exam Topic 1)
You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.
What are the two things you must do to achieve this? (Choose two.)

A. Provision new VMs using Terraform with the same VM names
B. Use the terraform import command for the existing VMs
C. Write Terraform configuration for the existing VMs
D. Run the terraform import-gcp command

Answer: BC

Explanation:
You should create the equivalent configuration first, and then run import to load it on the state file.

NEW QUESTION 17
- (Exam Topic 1)
Which of these options is the most secure place to store secrets foe connecting to a Terraform remote backend?

A. Defined in Environment variables
B. Inside the backend block within the Terraform configuration
C. Defined in a connection configuration outside of Terraform
D. None of above

Answer: A

Explanation:
https://www.terraform.io/language/settings/backends/configuration#credentials-and-sensitive-data Warning: We recommend using environment variables to supply credentials and other sensitive data. If you
use -backend-config or hardcode these values directly in your configuration, Terraform will include these values in both the .terraform subdirectory and in plan files. This can leak sensitive credentials.

NEW QUESTION 21
- (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 22
- (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 25
- (Exam Topic 1)
Which of the following is not a key principle of infrastructure as code?

A. Versioned infrastructure
B. Golden images
C. Idempotence
D. Self-describing infrastructure

Answer: B

Explanation:
Reference: https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code#:~:text=Idempotence%20is%20a%20principle%20of,of%20the%20environment's%20starting%20state.

NEW QUESTION 29
- (Exam Topic 1)
Terraform validate reports syntax check errors from which of the following scenarios?

A. Code contains tabs indentation instead of spaces
B. There is missing value for a variable
C. The state files does not match the current infrastructure
D. None of the above

Answer: B

Explanation:
The terraform validate command is used to validate the syntax of the terraform files. Terraform performs a syntax check on all the terraform files in the directory, and will display an error if any of the files doesn't validate. This command does not check formatting (e.g. tabs vs spaces, newlines, comments etc.). The following can be reported: invalid HCL syntax (e.g. missing trailing quote or equal sign) invalid HCL references (e.g. variable name or attribute which doesn't exist) same provider declared multiple times same module declared multiple times same resource declared multiple times invalid module name interpolation used in places where it's unsupported (e.g. variable, depends_on, module.source, provider) missing value for a variable (none of -var foo=... flag, -var-file=foo.vars flag, TF_VAR_foo environment variable, terraform.tfvars, or default value in the configuration) https://www.typeerror.org/docs/terraform/commands/validate
https://learning-ocean.com/tutorials/terraform/terraform-validate

NEW QUESTION 34
- (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 39
- (Exam Topic 1)
If a module uses a local variable, you can expose that value with a terraform output.

A. True
B. False

Answer: A

Explanation:
Output values are like function return values.
Reference: https://www.terraform.io/docs/language/values/locals.html https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION 44
- (Exam Topic 1)
Module variable assignments are inherited from the parent module and do not need to be explicitly set.

A. True
B. False

Answer: B

NEW QUESTION 48
- (Exam Topic 1)
You need to constrain the GitHub provider to version 2.1 or greater.
Which of the following should you put into the Terraform 0.12 configuration’s provider block?

A. version >= 2.1
B. version ~> 2.1
C. version = “<= 2.1”
D. version = “>= 2.1”

Answer: D

Explanation:
version = ">= 1.2.0, < 2.0.0"
A version constraint is a string literal containing one or more conditions, which are separated by commas. Each condition consists of an operator and a version number.
Version numbers should be a series of numbers separated by periods (like 1.2.0), optionally with a suffix to indicate a beta release.
The following operators are valid:
= (or no operator): Allows only one exact version number. Cannot be combined with other conditions.
!=: Excludes an exact version number.
>, >=, <, <=: Comparisons against a specified version, allowing versions for which the comparison is true. "Greater-than" requests newer versions, and "less-than" requests older versions.
~>: Allows only the rightmost version component to increment. For example, to allow new patch releases within a specific minor release, use the full version number: ~> 1.0.4 will allow installation of 1.0.5 and 1.0.10 but not 1.1.0. This is usually called the pessimistic constraint operator.
https://www.terraform.io/language/expressions/version-constraints

NEW QUESTION 49
- (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 52
- (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 56
- (Exam Topic 1)
You have declared an input variable called environment in your parent module. What must you do to pass the value to a child module in the configuration?

A. Add node_count = var.node_count
B. Declare the variable in a terraform.tfvars file
C. Declare a node_count input variable for child module
D. Nothing, child modules inherit variables of parent module

Answer: C

Explanation:
"That module may call other modules and connect them together by passing output values from one to input values of another."
https://www.terraform.io/language/modules/develop

NEW QUESTION 58
- (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 62
- (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 64
- (Exam Topic 1)
Setting the TF_LOG environment variable to DEBUG causes debug messages to be logged into syslog.

A. True
B. False

Answer: B

Explanation:
TF_LOG_PATH IS NOT REQUIRED, in the docs, they do not mention HAVE TO SET TF_LOG_PATH, it is optional, therefore without TF_LOG_PATH will cause detailed logs to appear on stderr.
https://www.computerhope.com/jargon/s/stderr.htm#:~:text=Stderr%2C%20also%20known%20as%20standard,

NEW QUESTION 67
- (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 71
- (Exam Topic 1)
Terraform and Terraform providers must use the same major version number in a single configuration.

A. True

B. False

Answer: B

Explanation:
https://www.terraform.io/language/expressions/version-constraints#terraform-core-and-provider-versions

NEW QUESTION 72
- (Exam Topic 1)
What does the default "local" Terraform backend store?

A. tfplan files
B. Terraform binary
C. Provider plugins
D. State file

Answer: D

Explanation:
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 76
- (Exam Topic 1)
Where in your Terraform configuration do you specify a state backend?

A. The terraform block
B. The resource block
C. The provider block
D. The datasource block

Answer: A

Explanation:
Backends are configured with a nested backend block within the top-level terraform block. Reference: https://www.terraform.io/docs/language/settings/backends/configuration.html https://www.terraform.io/language/settings/backends/configuration#using-a-backend-block

NEW QUESTION 79
- (Exam Topic 1)
What command does Terraform require the first time you run it within a configuration directory?

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

Answer: B

Explanation:
terraform init command is used to initialize a working directory containing Terraform configuration files. Reference: https://www.terraform.io/docs/cli/commands/init.html

NEW QUESTION 81
- (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 83
- (Exam Topic 1)
You have multiple team members collaborating on infrastructure as code (IaC) using Terraform, and want to apply formatting standards for readability.
How can you format Terraform HCL (HashiCorp Configuration Language) code according to standard Terraform style convention?

A. Run the terraform fmt command during the code linting phase of your CI/CD process
B. Designate one person in each team to review and format everyone's code
C. Manually apply two spaces indentation and align equal sign "=" characters in every Terraform file (*.tf)
D. Write a shell script to transform Terraform files using tools such as AWK, Python, and sed

Answer: A

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

NEW QUESTION 86
- (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 87
- (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 88
- (Exam Topic 2)
Which of the following represents a feature of Terraform Cloud that is NOT free to customers?

A. Roles and Team Management
B. WorkSpace Management
C. Private Module Registry
D. VCS Integration

Answer: A

Explanation:
Role Based Access Controls (RBAC) for controlling permissions for who has access to what configurations within an organization and it is not free to customers.
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 92
- (Exam Topic 2)
In regards to deploying resources in multi-cloud environments, what are some of the benefits of using Terraform rather than a provider's native tooling? (select three)

A. Terraform can help businesses deploy applications on multiple clouds and on-premises infrastructure.
B. Terraform is not cloud-agnostic and can be used to deploy resources across a single public cloud.
C. Terraform simplifies management and orchestration, helping operators build large-scale, multi-cloud infrastructure.
D. Terraform can manage cross-cloud dependencies.

Answer: ACD

Explanation:
Terraform is cloud-agnostic and allows a single configuration to be used to manage multiple providers, and to even handle cross-cloud dependencies. This simplifies management and orchestration, helping operators build large-scale multi-cloud infrastructures.
https://www.terraform.io/intro/use-cases.html

NEW QUESTION 93
- (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 94
- (Exam Topic 2)
John wants to use two different regions to deploy two different EC2 instances. He has specified two provider blocks in his providers.tf file.
provider "aws" { region = "us-east-1" } provider "aws" { region = "us-west-2" }
When he run terraform plan he encountered an error. How to fix this?

A. Use another provider version
B. Use alias for region = "us-west-2"
C. Use default keyword with region = "us-east-1"
D. It can not be fixed

Answer: B

NEW QUESTION 98
- (Exam Topic 2)
You want terraform plan and 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. This can be done using any of the local or remote backends
C. Remote Backends
D. Terraform Backends

Answer: C

Explanation:
The remote backend stores Terraform state and may be used to run operations in Terraform Cloud. 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. https://www.terraform.io/docs/backends/types/remote.html

NEW QUESTION 102
- (Exam Topic 2)
How can you ensure that the engineering team who has access to git repo will not create any non-compliant resources that might lead to a security audit failure in future. your team is using Hashicorp Terraform Enterprise Edition.

A. 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.
B. Implement a review process where every code will be reviewed before merging to the master branch.
C. 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.
D. 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.

Answer: C

Explanation:
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION 104
- (Exam Topic 2)
Which of the following clouds does not have a provider maintained HashiCorp?

A. IBM Cloud
B. DigitalOcean
C. OpenStack
D. AWS

Answer: A

Explanation:
IBM Cloud does not have a provider maintained by HashiCorp, although IBM Cloud does maintain their own Terraform provider.
https://www.terraform.io/docs/providers/index.html

NEW QUESTION 106
- (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 111
- (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 113
- (Exam Topic 2)
Terraform init can indeed be run only a few times, because, every time terraform init will initialize the project
, and download all plugins from the internet repository , regardless of whether they were present or not , and this increases the waiting time

A. True
B. False

Answer: B

Explanation:
Re-running init with modules already installed will install the sources for any modules that were added to configuration since the last init, but will not change any already-installed modules. Use -upgrade to override this behavior, updating all modules to the latest available source code.
https://www.terraform.io/docs/commands/init.html

NEW QUESTION 116
- (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 118
- (Exam Topic 2)
Workspaces in Terraform provides similar functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/docs/cloud/migrate/workspaces.html
Workspaces, managed with the terraform workspace command, aren't the same thing as Terraform Cloud's workspaces. Terraform Cloud workspaces act more like completely separate working directories; CLI workspaces are just alternate state files.

NEW QUESTION 122
- (Exam Topic 2)
terraform refresh will update the state file?

A. True
B. False

Answer: A

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.

NEW QUESTION 127
- (Exam Topic 2)
You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?

A. terraform unlock
B. terraform force-unlock
C. terraform state unlock
D. None of the above

Answer: B

Explanation:
 Command: force-unlock
Manually unlock the state for the defined configuration.
This will not modify your infrastructure. This command removes the lock on the state for the current configuration. The behavior of this lock is dependent on the backend being used. Local state files cannot be unlocked by another process.
https://www.terraform.io/docs/commands/force-unlock.html https://www.terraform.io/docs/state/locking.html
Terraform has a force-unlock command to manually unlock the state if unlocking failed.
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.

NEW QUESTION 130
- (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 133
- (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 138
- (Exam Topic 2)
Which of the following type of variable allows multiple values of several distinct types to be grouped together as a single value?

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

Answer: BC

Explanation:
Structural type of variable allows multiple values of several distinct types to be grouped together as a single value. They require a schema as an argument, to specify which types are allowed for which elements.
https://www.terraform.io/docs/configuration/types.html

NEW QUESTION 139
- (Exam Topic 2)
Which of the below terraform commands do not run terraform refresh implicitly before taking actual action of the command?

A. terraform apply
B. terraform destroy
C. terraform init
D. terraform import
E. terraform plan

Answer: CD

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

NEW QUESTION 143
- (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 148
- (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 150
- (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 152
- (Exam Topic 3)
Your manager has instructed you to start using terraform for the entire infra provisioning of the application stack. There are 4 environments – DEV , QA , UAT , and PROD. The application team has asked for complete segregation between these environments including the backend , state , and also configurations ,since there will be unique resources in different environments . What is the possible way to structure the terraform code to facilitate that.

A. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to a different backend.
B. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to the same backend.
C. Implement terraform workspaces , and map each environment with one workspace.
D. Enable remote backend storage . Configure 4 different backend storages , one for each environment.

Answer: A

Explanation:
In particular, 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

NEW QUESTION 154
- (Exam Topic 3)
A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip"} What is the best way to ensure that you are not using the latest version of the module?

A. Add a module version constraint in your configuration's backend block and specify a previous version.
B. Add a version key to the module configuration and specify a previous version.
C. Delete the latest version of the module in S3 to rollback to the previous version.
D. Add a version property to the module in Terraform's state file and specify a previous version.

Answer: C

Explanation:
Version constraints are supported only for modules installed from a module registry, such as the 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.
Only Terraform Registries support module versioning by using the version key, one cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version.
https://www.terraform.io/docs/configuration/modules.html#source

NEW QUESTION 158
- (Exam Topic 3)
In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

A. 5
B. 2
C. 3
D. 1

Answer: D

Explanation:
A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.
https://www.terraform.io/docs/cloud/workspaces/vcs.html

NEW QUESTION 160
- (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 164
- (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 167
- (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 170
- (Exam Topic 3)
You have created two workspaces PROD and DEV. You have switched to DEV and provisioned DEV infrastructure from this workspace. Where is your state file stored?

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

Answer: D

Explanation:
Terraform stores the workspace states in a directory called terraform.tfstate.d. This directory should be treated similarly to default workspace state file
terraform.tfstate main.tf
provider.tf terraform.tfstate.d DEV
terraform.tfstate # DEV workspace state file PROD
terraform.tfstate # PROD workspace state file terraform.tfvars # Default workspace state file variables.tf

NEW QUESTION 174
- (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 178
- (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 179
- (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 184
- (Exam Topic 3)
Dawn has created the below child module. Without changing the module, can she override the instance_type from t2.micro to t2.large form her code while calling this module?
* 1. resource "aws_instance" "myec2"
* 2. {
* 3. ami = "ami-082b5a644766e0e6f"
* 4. instance_type = "t2.micro
* 5. }

A. YES
B. No

Answer: B

Explanation:
As the instance_type is hard-coded in source module, you will not be able to change its value from destination module. Instead of hard-coding you should use variable with default values.

NEW QUESTION 185
- (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 189
- (Exam Topic 3)
After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?

A. ERROR
B. INFO
C. WARN
D. TRACE
E. DEBUG

Answer: D

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.
Examples:
export TF_LOG=DEBUG export TF_LOG=TRACE

NEW QUESTION 190
- (Exam Topic 3)
If you delete a remote backend from the configuration, will you need to rebuild your state files locally?

A. False
B. True

Answer: A

Explanation:
You can change your backend configuration at any time. You can change both the configuration itself as well as the type of backend (for example from "consul" to "s3").
Terraform will automatically detect any changes in your configuration and request a reinitialization. As part of the reinitialization process, Terraform will ask if you'd like to migrate your existing state to the new configuration. This allows you to easily switch from one backend to another.
https://www.terraform.io/docs/backends/config.html#changing-configuration

NEW QUESTION 193
- (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 195
- (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 200
- (Exam Topic 3)
Refer below code where pessimistic constraint operator has been used to specify a version of a provider. terraform { required_providers { aws = "~> 1.1.0" }}
Which of the following options are valid provider versions that satisfy the above constraint. (select two)

A. 1.1.1
B. 1.2.9
C. 1.1.8
D. 1.2.0

Answer: AC

Explanation:
Pessimistic constraint operator, constraining both the oldest and newest version allowed. For example, ~> 0.9 is equivalent to >= 0.9, < 1.0, and ~> 0.8.4, is equivalent to >= 0.8.4, < 0.9

NEW QUESTION 202
- (Exam Topic 3)
What happens when a terraform apply command is executed?

A. Creates the execution plan for the deployment of resources.
B. Applies the changes required in the target infrastructure in order to reach the desired configuration.
C. The backend is initialized and the working directory is prepped.
D. Reconciles the state Terraform knows about with the real-world infrastructure.

Answer: B

Explanation:
The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.
https://www.terraform.io/docs/commands/apply.html

NEW QUESTION 205
- (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 206
- (Exam Topic 3)
Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance. Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

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

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.
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 211
- (Exam Topic 3)
Which of the below command will upgrade the provider version to the latest acceptable one?

A. terraform plan upgrade
B. terraform provider -upgrade
C. terraform init -upgrade
D. terraform init -update

Answer: C

Explanation:
To upgrade to the latest acceptable version of each provider, run terraform init -upgrade. This command also upgrades to the latest versions of all Terraform modules.
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION 212
- (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 213
- (Exam Topic 3)
Which of the following variable definition files will terraform load automatically?

A. terraform.tfvar
B. Any files with names ending in .auto.tfvars.json
C. terraform.tfvars
D. terraform.tfvars.json

Answer: BCD

Explanation:
Terraform also automatically loads a number of variable definitions files if they are present: Files named exactly terraform.tfvars or terraform.tfvars.json.
Any files with names ending in .auto.tfvars or .auto.tfvars.json. https://www.terraform.io/docs/configuration/variables.html
https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files

NEW QUESTION 215
- (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 218
- (Exam Topic 3)
Taint the resource "aws_instance" "baz" resource that lives in module bar which lives in module foo.

A. terraform taint module.foo.module.bar.baz
B. terraform taint module.foo.bar.aws_instance.baz
C. terraform taint module.foo.module.bar.aws_instance.baz
D. terraform taint foo.bar.aws_instance.baz

Answer: C

Explanation:
Check resource addressing https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION 222
- (Exam Topic 3)
You cannot publish your own modules on the Terraform Registry.

A. False
B. True

Answer: A

Explanation:
https://www.terraform.io/docs/registry/modules/publish.html
You have a Terraform configuration file where a variable itemNum is defined as follows: variable "itemNum" { default = 3}

NEW QUESTION 224
- (Exam Topic 3)
You also have a defined the following environment variables in your shell: TF_itemNum =6, TF_VAR_itemNum =9. You also have a terraform.tfvars file with the following contents
itemNum = 7
When you run the following apply command, what is the value assigned to the itemNum variable? terraform apply -var itemNum =4

A. 10
B. 6
C. 1
D. 4
E. 3

Answer: D

Explanation:
The -var and -var-file methods of assigning variables have the highest precedence. https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION 227
- (Exam Topic 3)
You have created an AWS EC2 instance of type t2.micro through your terraform configuration file ec2.tf . Now you want to change the instance type from t2.micro to t2.medium. Accordingly you have changed your configuration file and and ran terraform plan. After running terraform plan you check the output and saw one instance will be updated from t2.micro --> t2.medium. After this you went to grab a coffee without running terraform apply and meanwhile a member of your team changed the instance type of that EC2 instance to t2.medium from aws console. After coming to your desk you run terraform apply. What will happen?

A. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
B. The instance type will be changed to t2.micro and again will be changed to t2.medium
C. terraform apply will through an error.
D. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.

Answer: A

NEW QUESTION 231
- (Exam Topic 3)
You can migrate the Terraform backend but only if there are no resources currently being managed.

A. False
B. True

Answer: A

Explanation:
If you need to migrate to another backend, such as Terraform Cloud, so you can continue managing it. By migrating your Terraform state, you can hand off infrastructure without de-provisioning anything.
https://www.terraform.io/docs/cloud/migrate/index.html

NEW QUESTION 232
- (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 233
- (Exam Topic 3)
Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?

A. Use data source aws_ami.
B. Create a map of region to ami id.
C. Create different configuration file for different region.
D. None of the above

Answer: A

Explanation:
https://www.terraform.io/docs/configuration/data-sources.html

NEW QUESTION 238
- (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 243
- (Exam Topic 3)
Mary has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code:
* 1. output "db_password"
* 2. {
* 3. value = local.db_password
* 4. }
Mary wants to hide the output value in the CLI after terraform apply? What is the best way?

A. Use secure parameter
B. Use sensitive parameter
C. Use cryptographic hash
D. Encrypt the value using encrypt() function

Answer: B

NEW QUESTION 248
- (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 252
- (Exam Topic 3)

A user has created three workspaces using the command line - prod, dev, and test. The user wants to create a fourth workspace named stage. Which command will the user execute to accomplish this?

A. terraform workspace new stage
B. terraform workspace -new stage
C. terraform workspace -create stage
D. terraform workspace create stage

Answer: A

Explanation:
The terraform workspace new command is used to create a new workspace. https://www.terraform.io/docs/commands/workspace/new.html

NEW QUESTION 257
- (Exam Topic 3)
Refer to the following terraform variable definition
variable "track_tag" { type = list default = ["data_ec2","integration_ec2","digital_ec2"]} track_tag = { Name = element(var.track_tag,count.index)}
If count.index is set to 2, which of the following values will be assigned to the name attribute of track_tag variable?

A. integration_ec2
B. digital_ec2
C. track_tag
D. data_ec2

Answer: B

NEW QUESTION 261
- (Exam Topic 4)
Which of the following is an invalid variable name?

A. count
B. web
C. var1
D. instance_name

Answer: A

Explanation:
https://www.terraform.io/intro/examples/count.html

NEW QUESTION 263
- (Exam Topic 4)
A Terraform output that sets the "sensitive" argument to true will not store that value in the state file.

A. True
B. False

Answer: B

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

NEW QUESTION 268
- (Exam Topic 4)
Using the terraform state rm command against a resource will destroy it.

A. True
B. False

Answer: B

NEW QUESTION 270
- (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 271
- (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 274
- (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

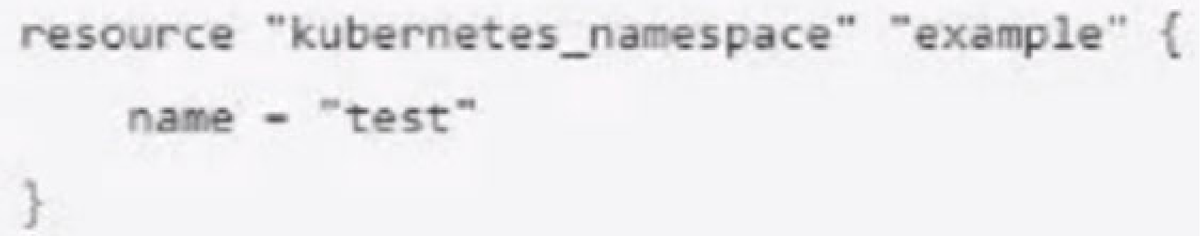
NEW QUESTION 275
- (Exam Topic 4)
What does the command terraform fmt do?

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 276
- (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 280
- (Exam Topic 4)
To check if all code in a Terraform configuration with multiple modules is properly formatted without making changes, what command should be run?

A. terraform fmt -check
B. terraform fmt -write-false
C. terraform fmt "list -recursive
D. terraform fmt -check -recursive

Answer: D

Explanation:
-check Check if the input is formatted. Exit status will be 0 if all input is properly formatted and non-zero otherwise.
-recursive Also process files in subdirectories. By default, only the given directory (or current directory) is processed.

NEW QUESTION 281
- (Exam Topic 4)
A user runs terraform init on their RHEL based server and per the output, two provider plugins are downloaded: $ terraform init
Initializing the backend... Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...
- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...
:
Terraform has been successfully initialized! Where are these plugins downloaded to?

A. The .terraform.plugins directory in the directory terraform init was executed in.
B. The .terraform/plugins directory in the directory terraform init was executed in.
C. /etc/terraform/plugins
D. The .terraform.d directory in the directory terraform init was executed in.

Answer: B

NEW QUESTION 284
- (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 288
- (Exam Topic 4)
Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 …) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region. What should you do? Your provider setting is as below
# The default provider configuration provider "aws" { region = "us-east-1" }

A. No way to enable this via a single script . Write 2 different scripts with different default providers in the 2 scripts , one for us-east , another for us-west.
B. Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.
C. Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).
D. Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.

Answer: C

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"
}
```

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

NEW QUESTION 292
- (Exam Topic 4)
Which one is the right way to import a local module names consul?

A. module "consul" { source = "consul"}
B. module "consul" { source = "./consul"}
C. module "consul" { source = "../consul"}
D. module "consul" { source = "module/consul"}

Answer: BC

Explanation:
A local path must begin with either ./ or ../ to indicate that a local path is intended, to distinguish from a module registry address.
module "consul" {
source = "./consul"
}

NEW QUESTION 293
- (Exam Topic 4)
You have configured an Auto Scaling group in AWS to automatically scale the number of instances behind a load balancer based on the instances CPU utilization. The instances are configured using a Launch Configuration. You have observed that the Auto Scaling group doesn't successfully scale when you apply changes that require replacing the Launch Configuration. Why is this happening?

A. You need to configure an explicit dependency for the Auto Scaling group using the depends_on meta-parameter.
B. You need to configure an explicit dependency for the Launch Configuration using the depends_on meta-parameter.
C. You need to configure the Auto Scaling group's create_before_destroy meta-parameter.
D. You need to configure the Launch Configuration's create_before_destroy meta-parameter.

Answer: D

Explanation:
https://www.terraform.io/docs/providers/aws/r/launch_configuration.html#using-withautoscaling-groups

NEW QUESTION 294
- (Exam Topic 4)
Terra form installs its providers during which phase?

A. Man
B. Init
C. Refresh
D. All of the above

Answer: B

Explanation:
Providers are installed in the init phase

NEW QUESTION 295
- (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

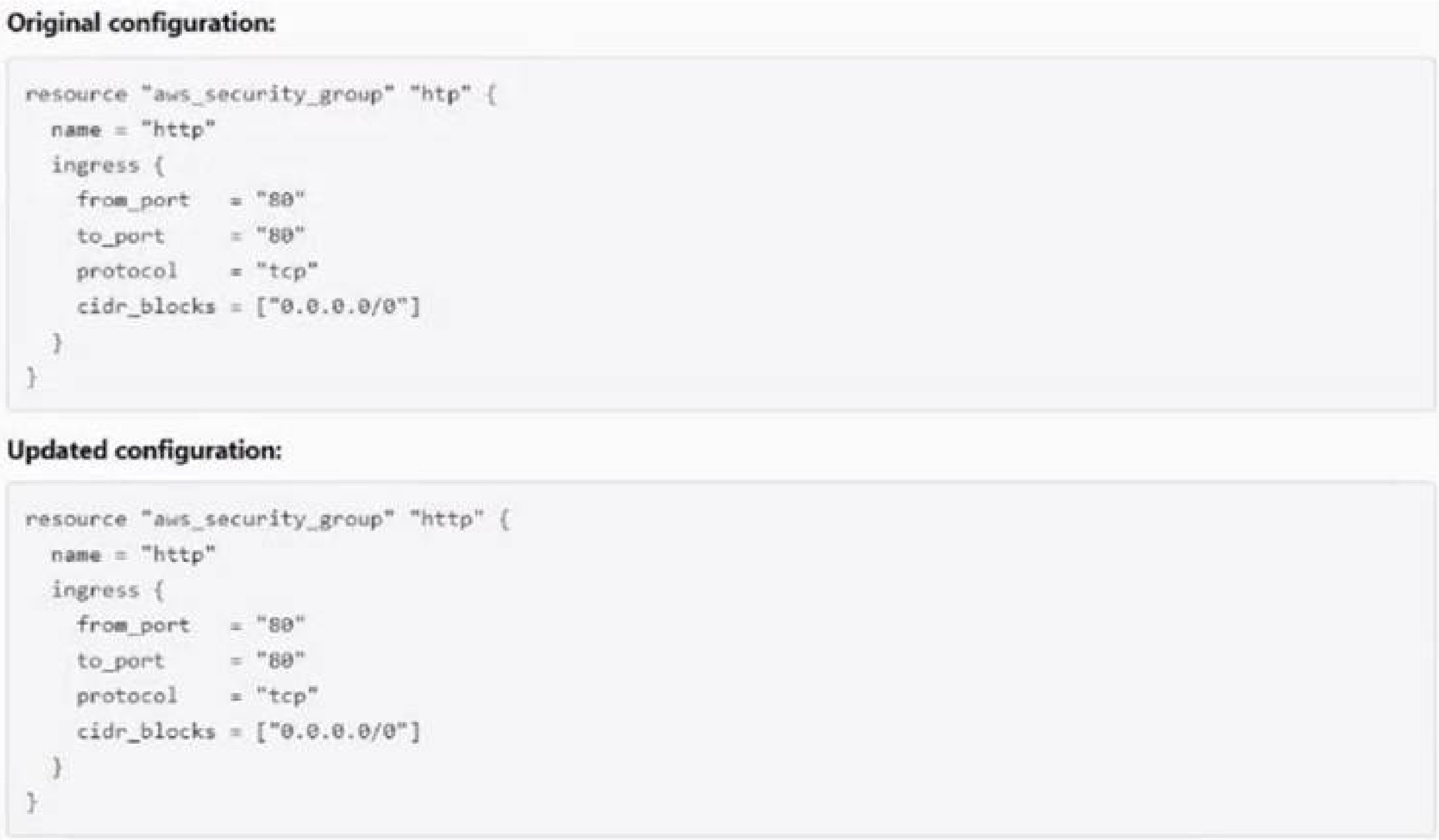
NEW QUESTION 300
- (Exam Topic 4)
John is writing a module and within the module, there are multiple places where he has to use the same conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration,. What is a better approach to dealing with this?

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

Answer: A

Explanation:
A local value assigns a name to an expression, allowing it to be used multiple times within a module without repeating it.
https://www.terraform.io/docs/configuration/locals.html

NEW QUESTION 301
- (Exam Topic 4)
You have modified your Terraform configuration to fix a typo in the Terraform ID of a resource from aws_security_group.http to aws_security_group.http



Original configuration:

```
resource "aws_security_group" "htp" {
  name = "http"
  ingress {
    from_port   = "80"
    to_port     = "80"
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Updated configuration:

```
resource "aws_security_group" "http" {
  name = "http"
  ingress {
    from_port   = "80"
    to_port     = "80"
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Which of the following commands would you run to update the ID in state without destroying the resource?

A. terraform refresh
B. terraform apply
C. terraform mv aws-security-group.htp aws-security-group.http

Answer: C

Explanation:
The terraform state mv command changes which resource address in your configuration is associated with a particular real-world object. Use this to preserve an object when renaming a resource, or when moving a resource into or out of a child module.

NEW QUESTION 305
- (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 306
- (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 310
- (Exam Topic 4)
Given the below resource configuration - resource "aws_instance" "web" { # ... count = 4 }
What does the terraform resource address aws_instance.web refer to?

A. It refers to all 4 web instances , together , for further individual segregation , indexing is required , with a 0 based index.
B. It refers to the last web EC2 instance , as by default , if no index is provided , the last / N-1 index is used.
C. It refers to the first web EC2 instance out of the 4 ,as by default , if no index is provided , the first / 0th index is used.
D. The above will result in a syntax error , as it is not syntactically correct . Resources defined using count , can only be referenced using indexes.

Answer: A

Explanation:
A Resource Address is a string that references a specific resource in a larger infrastructure. An address is made up of two parts:
[module path][resource spec] Module path:

A module path addresses a module within the tree of modules. It takes the form: module.A.module.B.module.C...
Multiple modules in a path indicate nesting. If a module path is specified without a resource spec, the address applies to every resource within the module. If the module path is omitted, this addresses the root module.
Given a Terraform config that includes: resource "aws_instance" "web" {
# ...
count = 4
}
An address like this: aws_instance.web[3]
Refers to only the last instance in the config, and an address like this: aws_instance.web
Refers to all four "web" instances. https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION 312
- (Exam Topic 4)
True or False? terraform init cannot automatically download Community providers.

A. False
B. True

Answer: B

NEW QUESTION 315
- (Exam Topic 4)
Which task does teraform init not perform?

A. Sources any modules and copies the configuration locally
B. Validates all required variables are present
C. Connects to the backend
D. Sources all providers present in the configuration and ensures they are downloaded and available locally

Answer: B

NEW QUESTION 317
- (Exam Topic 4)
The Terraform CLI will print output values from a child module after running terraform apply.

A. True
B. False

Answer: A

NEW QUESTION 320
- (Exam Topic 4)
You're preparing to install Terraform on client workstations and want to see which operating systems are supported. Which of the following operating systems is supported?

A. Windows
B. Amazon Linux
C. FreeBSD
D. Solaris
E. MacOS
F. All of the above

Answer: F

NEW QUESTION 322
- (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 325
- (Exam Topic 4)
When should Terraform configuration files be written when running terraform import on existing infrastructure?

A. Infrastructure can be imported without corresponding Terraform code
B. Terraform will generate the corresponding configuration files for you
C. You should write Terraform configuration files after the next terraform import is executed
D. Terraform configuration should be written before terraform import is executed

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.
Source: https://www.terraform.io/cli/import

NEW QUESTION 328
- (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 329
- (Exam Topic 4)
terraform destroy is the only way to remove infrastructure.

A. True
B. False

Answer: B

NEW QUESTION 330
- (Exam Topic 4)
Resources in terraform can have same identifiers(Resource type + Block name).

A. True
B. False

Answer: B

NEW QUESTION 334
- (Exam Topic 4)

What is the result of the following terraform function call?

A. hello
B. what?
C. goodbye

Answer: B

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

NEW QUESTION 339
- (Exam Topic 4)
True or False: Workspaces provide identical functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

A. True
B. False

Answer: B

Explanation:

https://www.terraform.io/docs/cloud/workspaces/index.html https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION 342
- (Exam Topic 4)
You have created a custom variable definition file my_vars.tfvars. How will you use it for provisioning infrastructure?

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

Answer: D

Explanation:
To set lots of variables, it is more convenient to specify their values in a variable definitions file (with a filename ending in either .tfvars or .tfvars.json) and then specify that file on the command line with -var-file:
terraform apply -var-file="my_vars.tfvars" https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files

NEW QUESTION 345
- (Exam Topic 4)
True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

A. False
B. True

Answer: B

Explanation:
The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

NEW QUESTION 347
- (Exam Topic 4)
Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?

A. Use terraform console command to have an interactive UI with full access to the underlying terraform state to run your interpolations , and debug at real-time.
B. Add a breakpoint in your code, using the watch keyword , and output the value to console for temporary debugging.
C. Use terraform zipmap function , it will be able to easily do the interpolations without complex code.
D. Use terraform console command to have an interactive UI , but you can only use it with local state , and it does not work with remote state.

Answer: A

Explanation:
The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.
https://www.terraform.io/docs/commands/console.html

NEW QUESTION 351
- (Exam Topic 4)
In a Terraform Cloud workspace linked to a version control repository, speculative plan runs start automatically when you merge or commit changes to version control.

A. True

B. False

Answer: B

NEW QUESTION 354
- (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 359
- (Exam Topic 4)
Your company has a lot of workloads in AWS , and Azure that were respectively created using CloudFormation , and AzureRM Templates. However , now your CIO has decided to use Terraform for all new projects , and has asked you to check how to integrate the existing environment with terraform code. What should be your next plan of action?

A. Tell the CIO that this is not possible . Resources created in CloudFormation , and AzureRM templates cannot be tracked using terraform.
B. Use terraform import command to import each resource one by one .
C. This is only possible in Terraform Enterprise , which has the TerraformConverter exe that can take any other template language like AzureRM and convert to Terraform code.
D. Just write the terraform config file for the new resources , and run terraform apply , the state file will automatically be updated with the details of the new resources to be imported.

Answer: B

NEW QUESTION 362
- (Exam Topic 4)
A single terraform resource file that defines an aws_instance resource can simple be renamed to azurerm_virtual_machine in order to switch cloud providers

A. True
B. False

Answer: B

Explanation:
Providers usually require some configuration of their own to specify endpoint URLs, regions, authentication settings.
Providers Initialization can be done by either explicitly via a provider block or by adding a resource from that provide
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION 365
- (Exam Topic 4)
A terraform apply can not ______ infrastructure.

A. import
B. provision
C. destroy
D. change

Answer: A

NEW QUESTION 368
- (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 371
- (Exam Topic 4)
If a Terraform creation-time provisioner fails, what will occur by default?

A. The resource will not be affected, but the provisioner will need to be applied again
B. The resource will be destroyed

C. The resource will be marked as "tainted"
D. Nothing, provisioners will not show errors in the command line

Answer: C

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 .

NEW QUESTION 376
- (Exam Topic 4)
In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?

A. Resource dependencies are identified and maintained in a file called resource.dependencie
B. Each terraform provider is required to maintain a list of all resource dependencies for the provider and it's included with the plugin during initialization when terraform init is execute
C. The file is located in the terraform.d folder.
D. The terraform binary contains a built-in reference map of all defined Terraform resource dependencies.Updates to this dependency map are reflected in terraform version
E. To ensure you are working with thelatest resource dependency map you much be running the latest version of Terraform.
F. Resource dependencies are handled automatically by the depends_on meta_argument, which is set to true by default.
G. Terraform analyses any expressions within a resource block to find references to other objects, and treats those references as implicit ordering requirements when creating, updating, or destroying resources.

Answer: D

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

NEW QUESTION 379
- (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 381
- (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 382
- (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 383
- (Exam Topic 4)
Select the feature below that best completes the sentence:
The following list represents the different types of ______ available in Terraform.
* 1. max
* 2. min
* 3. join
* 4. replace
* 5. list
* 6. length
* 7. range

A. Backends
B. Data sources
C. Named values
D. Functions

Answer: D

Explanation:
The Terraform language includes a number of built-in functions that you can call from within expressions to transform and combine values. The Terraform language does not support user-defined functions, and only the functions built into the language are available for use.
https://www.terraform.io/docs/configuration/functions.html

NEW QUESTION 387
- (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 391
- (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 394
- (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 396
- (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 399
- (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 400
- (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 404
- (Exam Topic 4)
You want to share Terraform state with your team, store it securely and provide state locking. How would you do this? Choose three correct answers.

A. Using the consul Terraform backend.
B. Using the remote Terraform backend with Terraform Cloud / Terraform Enterprise.
C. Using the local backend.
D. Using the s3 terraform backen
E. The dynamodb_field option e not needed.
F. Using an s3 terraform backend with an appropriate IAM policy and dynamodb_field option configured.

Answer: ABE

NEW QUESTION 407
- (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 410
- (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 413
- (Exam Topic 4)
Which of the following is true about terraform apply? (Choose two.)

A. It only operates on infrastructure defined in the current working directory or workspace
B. You must pass the output of a terraform plan command to it
C. Depending on provider specification, Terraform may need to destroy and recreate your infrastructure resources
D. By default, it does not refresh your state file to reflect current infrastructure configuration
E. You cannot target specific resources for the operation

Answer: AC

Explanation:
https://www.terraform.io/cli/run

NEW QUESTION 415
- (Exam Topic 4)
Which provider authentication method prevents credentials from being stored in the state file?

A. Using environment variables
B. Specifying the login credentials in the provider block
C. Setting credentials as Terraform variables
D. None of the above

Answer: A

NEW QUESTION 419
- (Exam Topic 4)
You want to define multiple data disks as nested blocks inside the resource block for a virtual machine. What Terraform feature would help you define the blocks using the values in a variable?

A. Local values
B. Dynamic blocks

C. Count arguments
D. Collection functions

Answer: B

NEW QUESTION 424
- (Exam Topic 4)
What advantage does an operations team that uses infrastructure as code have?

A. The ability to delete infrastructure
B. The ability to reuse best practice configurations and settings
C. The ability to autoscale a group of servers
D. The ability to update existing infrastructure

Answer: B

NEW QUESTION 427
- (Exam Topic 4)
Which of the following does terraform apply change after you approve the execution plan? Choose two correct answers.

A. The execution plan
B. Terraform code
C. Cloud infrastructure
D. State file
E. The .terraform directory

Answer: CD

NEW QUESTION 428
- (Exam Topic 4)
A module can always refer to all variables declared in its parent module.

A. True
B. False

Answer: B

Explanation:
Modules do not inherit variables from the parent module. All modules are self-contained units. So you have to explicitly define variables in the child module, and then explicit set these variables in the parent module, when you instantiate the child module.

NEW QUESTION 433
- (Exam Topic 4)
A junior admin accidentally deleted some of your cloud instances. What does Terraform do when you run terraform apply?

A. Build a completely brand new set of infrastructure
B. Tear down the entire workspace infrastructure and rebuild it
C. Rebuild only the instances that were deleted Most Voted
D. Stop and generate an error message about the missing instances

Answer: C

NEW QUESTION 437
- (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 441
- (Exam Topic 4)
Which is the best way to specify a tag of v1.0.0 when referencing a module stored in Git (for example git::https://example.com/vpc.git)?

A. Append ref=v1. 0. 0 argument to the source path Most Voted
B. Add version = "1.0.0" parameter to module block
C. Nothing " modules stored on GitHub always default to version 1.0.0
D. Modules stored on GitHub do not support versioning

Answer: A

Explanation:

https://www.terraform.io/language/modules/sources#selecting-a-revision

NEW QUESTION 445
- (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 448
- (Exam Topic 4)
You have a Terraform configuration that defines a single virtual machine with no references to it. You have run terraform apply to create the resource, and then removed the resource definition from your Terraform configuration file.
What will happen when you run terraform apply in the working directory again?

A. Nothing
B. Terraform will destroy the virtual machine
C. Terraform will error
D. Terraform will remove the virtual machine from the state file, but the resource will still exist

Answer: B

Explanation:
If you remove the resource from your config file and the resource is in your state file, terraform will apply the configuration in the config file - which is to delete the resource

NEW QUESTION 453
- (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 458
- (Exam Topic 4)
What are some of the features of Terraform state? (select three)

A. inspection of cloud resources
B. determining the correct order to destroy resources
C. mapping configuration to real-world resources
D. increased performance

Answer: CD

NEW QUESTION 463
- (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.

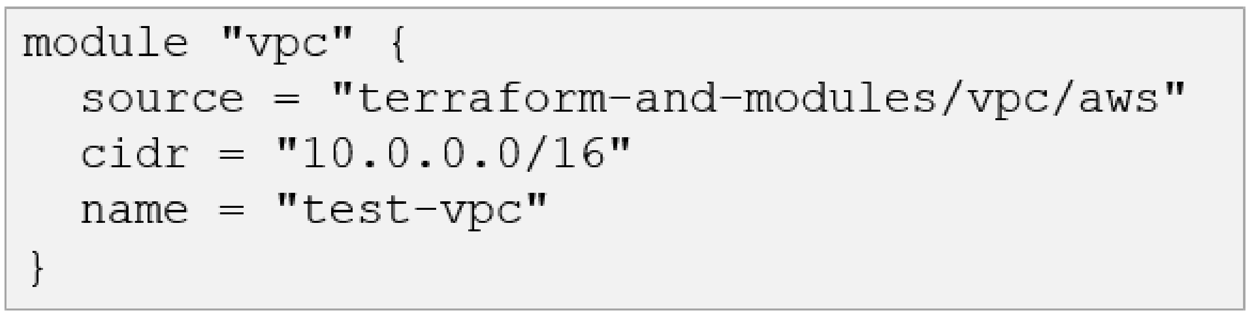
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION 464
- (Exam Topic 4)
Choose the best option from below to make Terraform code more user configuration-centric.

A. Variables
B. Local values
C. Input Variable
D. Modules

Answer: C

NEW QUESTION 468
- (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 472
- (Exam Topic 4)
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.

A. False
B. True

Answer: B

NEW QUESTION 477
- (Exam Topic 4)
As a member of the operations team, you need to run a script on a virtual machine created by Terraform. Which provisioned is best to use in your Terraform code?

A. local-exec
B. file
C. null-exec
D. remote-exec

Answer: D

Explanation:
https://www.terraform.io/language/resources/provisioners/remote-exec

NEW QUESTION 480
- (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 485
- (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 486
- (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 488
- (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 490
- (Exam Topic 4)
Which of the below backends support state locking?

A. S3
B. consul
C. azurerm
D. artifactory

Answer: ABC

NEW QUESTION 491
- (Exam Topic 4)
As a member of an operations team that uses infrastructure as code (laC) practices, you are tasked with making a change to an infrastructure stack running in a public cloud. Which pattern would follow laC best practices for making a change?

A. Make the change via the public cloud API endpoint
B. Make the change programmatically via the public cloud CLI
C. Submit a pull request and wait for an approved merge of the proposed changes
D. Use the public cloud console to make the change after a database record has been approved
E. Clone the repository containing your infrastructure code and then run the code

Answer: C

NEW QUESTION 492
- (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 496

- (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 501
......