# Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

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

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

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

**Answer:** C

**Explanation:**

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

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

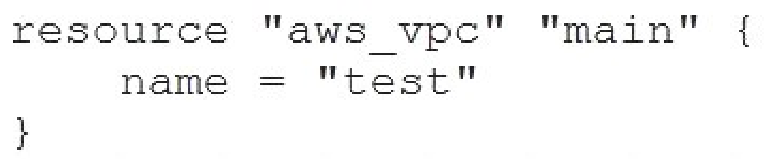
A. Secure variable storage
B. Support for multiple cloud providers
C. Dry runs with terraform plan
D. Using the workspace as a data source

**Answer:** A

**Explanation:**

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

**NEW QUESTION 3**
- (Exam Topic 1)
What is the provider for this fictitious resource?



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

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

**Answer:** C

**Explanation:**

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

**NEW QUESTION 4**
- (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 5**
- (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

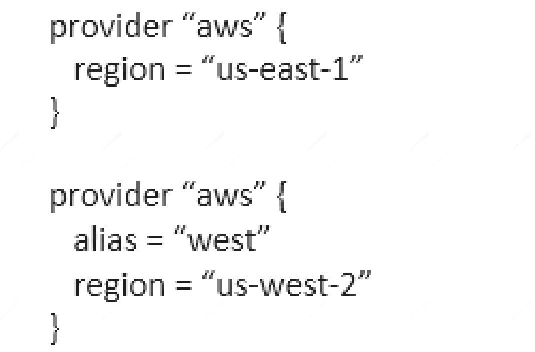
**NEW QUESTION 6**
- (Exam Topic 1)
When should you use the force-unlock command?

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

**Answer:** C

**Explanation:**
Be very careful with this command. If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed. Source: https://www.terraform.io/language/state/locking
https://www.terraform.io/cli/commands/force-unlock

**NEW QUESTION 7**
- (Exam Topic 1)
You need to deploy resources into two different cloud regions in the same Terraform configuration. To do that, you declare multiple provider configurations as follows:



```
provider “aws” {
  region = “us-east-1”
}

provider “aws” {
  alias = “west”
  region = “us-west-2”
}
```

What meta-argument do you need to configure in a resource block to deploy the resource to the “us-west-2” AWS region?

A. alias = west
B. provider = west
C. provider = aws.west
D. alias = aws.west

**Answer:** C

**Explanation:**

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

**NEW QUESTION 8**
- (Exam Topic 1)
You have used Terraform to create an ephemeral development environment in the cloud and are now ready to destroy all the infrastructure described by your Terraform configuration. To be safe, you would like to first see all the infrastructure that will be deleted by Terraform.
Which command should you use to show all of the resources that will be deleted? (Choose two.)

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

**Answer:** AD

**Explanation:**

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

**NEW QUESTION 9**
- (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 10**
- (Exam Topic 1)
Which task does terraform init not perform?

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

**Answer:** D

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

**NEW QUESTION 10**
- (Exam Topic 1)
Which statement describes a goal of infrastructure as code?

A. An abstraction from vendor specific APIs
B. Write once, run anywhere
C. A pipeline process to test and deliver software
D. The programmatic configuration of resources

**Answer:** D

**Explanation:**
The purpose of infrastructure as code is to enable developers or operations teams to automatically manage, monitor and provision resources, rather than manually configure discrete hardware devices and operating systems. Infrastructure as code is sometimes referred to as programmable or software-defined infrastructure.

**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 15**
- (Exam Topic 1)
You just scaled your VM infrastructure and realized you set the count variable to the wrong value. You correct the value and save your change.
What do you do next to make your infrastructure match your configuration?

A. Run an apply and confirm the planned changes
B. Inspect your Terraform state because you want to change it
C. Reinitialize because your configuration has changed
D. Inspect all Terraform outputs to make sure they are correct

**Answer:** A

**NEW QUESTION 17**
- (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 22**
- (Exam Topic 1)
What is the workflow for deploying new infrastructure with Terraform?

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

D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

**Answer:** D

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

**NEW QUESTION 24**
- (Exam Topic 1)
Terraform variables and outputs that set the "description" argument will store that description in the state file.

A. True
B. False

**Answer:** B

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

**NEW QUESTION 25**
- (Exam Topic 1)
Terraform requires the Go runtime as a prerequisite for installation.

A. True
B. False

**Answer:** B

**Explanation:**
https://www.terraform.io/plugin/sdkv2/guides/v1-upgrade-guide and https://www.terraform.io/plugin/sdkv2/guides/v2-upgrade-guide

**NEW QUESTION 26**
- (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 27**
- (Exam Topic 1)
When using Terraform to deploy resources into Azure, which scenarios are true regarding state files? (Choose two.)

A. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in a new state file
B. When a change is made to the resources via the Azure Cloud Console, Terraform will update the state file to reflect them during the next plan or apply
C. When a change is made to the resources via the Azure Cloud Console, the current state file will not be updated
D. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in the current state file

**Answer:** BC

**NEW QUESTION 29**
- (Exam Topic 1)
A Terraform provider is not responsible for:

A. Understanding API interactions with some service
B. Provisioning infrastructure in multiple clouds
C. Exposing resources and data sources based on an API
D. Managing actions to take based on resource differences

**Answer:** B

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

**NEW QUESTION 34**
- (Exam Topic 1)
How can you trigger a run in a Terraform Cloud workspace that is connected to a Version Control System (VCS) repository?

A. Only Terraform Cloud organization owners can set workspace variables on VCS connected workspaces
B. Commit a change to the VCS working directory and branch that the Terraform Cloud workspace is connected to
C. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces
D. Only Terraform Cloud organization owners can approve plans in VCS connected workspaces

**Answer:** B

**Explanation:**
"In a workspace linked to a VCS repository, runs start automatically when you merge or commit changes to version control.
A workspace is linked to one branch of a VCS repository and ignores changes to other branches. You can specify which files and directories within your repository trigger runs. "
https://www.terraform.io/cloud-docs/run/ui#automatically-starting-runs

**NEW QUESTION 39**
- (Exam Topic 1)
Which of the following is not an action performed by terraform init?

A. Create a sample main.tf file
B. Initialize a configured backend
C. Retrieve the source code for all referenced modules
D. Load required provider plugins

**Answer:** A

**NEW QUESTION 44**
- (Exam Topic 1)
How is the Terraform remote backend different than other state backends such as S3, Consul, etc.?

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

**Answer:** A

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

**NEW QUESTION 47**
- (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 51**
- (Exam Topic 2)
True or False: A list(...) contain a number of values of the same type while an object(...) can contain a number of values of different types.

A. False
B. True

**Answer:** B

**Explanation:**
Collection Types
A collection type allows multiple values of one other type to be grouped together as a single value. The type of value within a collection is called its element type. All collection types must have an element type, which is provided as the argument to their constructor.
For example, the type list(string) means "list of strings", which is a different type than list(number), a list of numbers. All elements of a collection must always be of the same type.
The three kinds of collection type in the Terraform language are:
* list(...): a sequence of values identified by consecutive whole numbers starting with zero.
The keyword list is a shorthand for list(any), which accepts any element type as long as every element is the same type. This is for compatibility with older configurations; for new code, we recommend using the full form.
* map(...): a collection of values where each is identified by a string label.
The keyword map is a shorthand for map(any), which accepts any element type as long as every element is the same type. This is for compatibility with older

configurations; for new code, we recommend using the full form.
* set(...): a collection of unique values that do not have any secondary identifiers or ordering. https://www.terraform.io/docs/configuration/types.html
Structural Types
A structural type allows multiple values of several distinct types to be grouped together as a single value. Structural types require a schema as an argument, to specify which types are allowed for which elements.
The two kinds of structural type in the Terraform language are:
* object(...): a collection of named attributes that each have their own type.
The schema for object types is { <KEY> = <TYPE>, <KEY> = <TYPE>, ... } — a pair of curly braces containing a comma-separated series of <KEY> = <TYPE> pairs. Values that match the object type must contain all of the specified keys, and the value for each key must match its specified type. (Values with additional keys can still match an object type, but the extra attributes are discarded during type conversion.)
* tuple(...): a sequence of elements identified by consecutive whole numbers starting with zero, where each element has its own type.
The schema for tuple types is [<TYPE>, <TYPE>, ...] — a pair of square brackets containing a comma-separated series of types. Values that match the tuple type must have exactly the same number of elements (no more and no fewer), and the value in each position must match the specified type for that position.
For example: an object type of object({ name=string, age=number }) would match a value like the following:
{
name = "John" age = 52
}
Also, an object type of object({ id=string, cidr_block=string }) would match the object produced by a reference to an aws_vpc resource, like aws_vpc.example_vpc; although the resource has additional attributes, they would be discarded during type conversion.
Finally, a tuple type of tuple([string, number, bool]) would match a value like the following: ["a", 15, true]
https://www.terraform.io/docs/configuration/types.html

**NEW QUESTION 55**
- (Exam Topic 2)
The Terraform language does not support user-defined functions, and so only the functions built in to the language are available for use.

A. False
B. True

**Answer:** B

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

**NEW QUESTION 60**
- (Exam Topic 2)
Environment variables can be used to set variables. The environment variables must be in the format " "_<variablename>. Select the correct prefix string from the following list.

A. TF_CLI_ARGS
B. TF_VAR
C. TF_VAR_
D. TF_VAR_ENV

**Answer:** C

**Explanation:**
Environment variables can be used to set variables. The environment variables must be in the format TF_VAR_name and this will be checked last for a value. For example:
export TF_VAR_region=us-west-1
export TF_VAR_ami=ami-049d8641 export TF_VAR_alist='[1,2,3]'
export TF_VAR_amap='{ foo = "bar", baz = "qux" }' https://www.terraform.io/docs/commands/environment-variables.html

**NEW QUESTION 62**
- (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 64**
- (Exam Topic 2)
If you enable TF_LOG = DEBUG, the log will be stored in syslog.log file in the currect directory.

A. False
B. True

**Answer:** A

**Explanation:**
https://www.terraform.io/docs/internals/debugging.html

**NEW QUESTION 69**
- (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 72**
- (Exam Topic 2)
Which Terraform command will force a marked resource to be destroyed and recreated on the next apply?

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

**Answer:** C

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

**NEW QUESTION 74**
- (Exam Topic 2)
terraform 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 79**
- (Exam Topic 2)
Which of the following best describes a Terraform provider?

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

**Answer:** A

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

**NEW QUESTION 81**
- (Exam Topic 2)
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 84**
- (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 89**
- (Exam Topic 2)
What is the command you can use to set an environment variable named "var1"of type String?

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

**Answer:** D

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

**NEW QUESTION 92**
- (Exam Topic 2)
What is the purpose of using the local-exec provisioner? (Select Two)

A. To invoke a local executable.
B. Executes a command on the resource to invoke an update to the Terraform state.
C. To execute one or more commands on the machine running Terraform.
D. Ensures that the resource is only executed in the local infrastructure where Terraform is deployed.

**Answer:** AC

**Explanation:**
The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource. Note that even though the resource will be fully created when the provisioner is run, there is no guarantee that it will be in an operable state - for example system services such as sshd may not be started yet on compute resources.
Example usage

```
resource "aws_instance" "web" {
# ...
provisioner "local-exec" {
command = "echo ${aws_instance.web.private_ip} >> private_ips.txt"
}
}
```

Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives.
https://www.terraform.io/docs/provisioners/local-exec.html

**NEW QUESTION 95**
- (Exam Topic 3)
Which of the following state management command allow you to retrieve a list of resources that are part of the state file?

A. terraform state list
B. terraform state view
C. terraform view
D. terraform list

**Answer:** A

**Explanation:**
The terraform state list command is used to list resources within a Terraform state. Usage: terraform state list [options] [address...]
The command will list all resources in the state file matching the given addresses (if any). If no addresses are given, all resources are listed.
https://www.terraform.io/docs/commands/state/list.html

**NEW QUESTION 97**
- (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 100**
- (Exam Topic 3)
Which of the below options is a valid interpolation syntax for retrieving a data source?

A. ${google_storage_bucket.backend}
B. ${azurerm_resource_group.test.data}
C. ${aws_instance.web.id.data}
D. ${data.google_dns_keys.foo_dns_keys.key_signing_keys[0].ds_record}

**Answer:** D

**Explanation:**
Data source attributes are interpolated with the general syntax data.TYPE.NAME.ATTRIBUTE. The interpolation for a resource is the same but without the data. prefix (TYPE.NAME.ATTRIBUTE).
https://www.terraform.io/docs/configuration-0-11/interpolation.html#attributes-of-a-data-source

**NEW QUESTION 102**
- (Exam Topic 3)
Your manager has instructed you to start using terraform for your day-to-day operations, but your security team is concerned about the terraform state files. They have heard it contains confidential information, and are worried that it will not be securely protected. What should be your response to the security team in this regard?

A. Inform the security team that using terraform state is optional . There are ways to avoid it , and you will do the same.
B. Ensure that the state is managed in a remote backend , preferably an enterprise grade state management system like Terraform Cloud.
C. Mask the confidential entries in the terraform state file , using Vault Enterprise, another Hashicorp product , while keeping it locally.
D. Keep the state file locally on each developer machine , and ensure that there is a local protection software like KeyPass protecting it.

**Answer:** B

**Explanation:**
https://www.terraform.io/docs/state/index.html
State is very important topic for exam. Please read all of the below subtopics Purpose
Import Existing Resources Locking
Workspaces Remote State Sensitive Data

**NEW QUESTION 105**
- (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 109**
- (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 112**

- (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 113**
- (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 117**
- (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 121**
- (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 126**
- (Exam Topic 3)
A data block requests that Terraform read from a given data source and export the result under the given local name.

A. False
B. True

**Answer:** B

**NEW QUESTION 131**
- (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 133**
- (Exam Topic 3)
Command terraform refresh will update state file?

A. False
B. True

**Answer:** B

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

**NEW QUESTION 134**
- (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 139**
- (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 142**
- (Exam Topic 3)
The terraform state command can be used to _____

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

**Answer:** A

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

**NEW QUESTION 147**
- (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 151**
- (Exam Topic 3)
By default, provisioners that fail will also cause the Terraform apply itself to error. How can you change this default behavior within a provisioner?

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

**Answer:** C

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

**NEW QUESTION 155**
- (Exam Topic 3)
Multiple configurations for the same provider can be used in a single configuration file.

A. False
B. True

**Answer:** B

**Explanation:**
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

```
# The default provider configuration provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region provider "aws" {
alias = "west" region = "us-west-2"
}
```

The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances

**NEW QUESTION 158**
- (Exam Topic 3)
During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

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

**Answer:** A

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

**NEW QUESTION 162**
- (Exam Topic 4)
Which of the following is an invalid variable name?

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

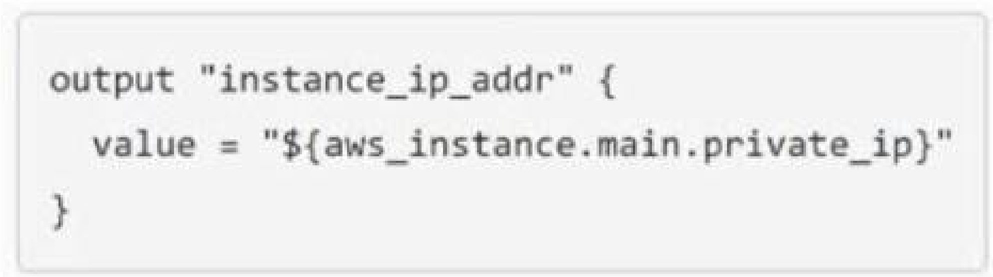
**Answer:** A

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

**NEW QUESTION 166**
- (Exam Topic 4)
You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name

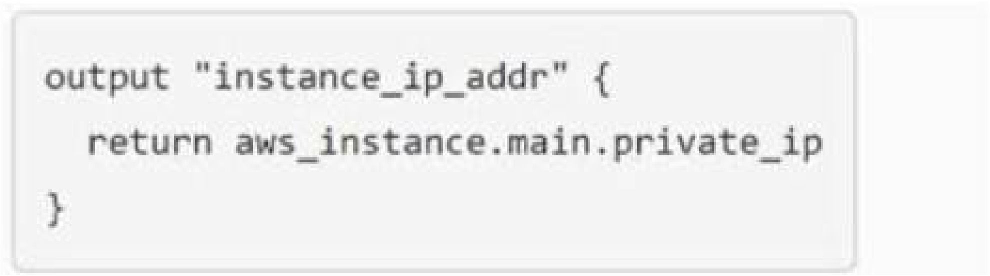
the instance resource "main".
Which of these is the correct way to define the output value using HCL2?

A.



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

B.



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

A. Option A
B. Option B

**Answer:** A

**NEW QUESTION 167**
- (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 171**
- (Exam Topic 4)
True or False. The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. If drift is detected between the real-world infrastructure and the last known-state, it will modify the infrastructure to correct the drift.

A. False
B. True

**Answer:** A

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

**NEW QUESTION 174**
- (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

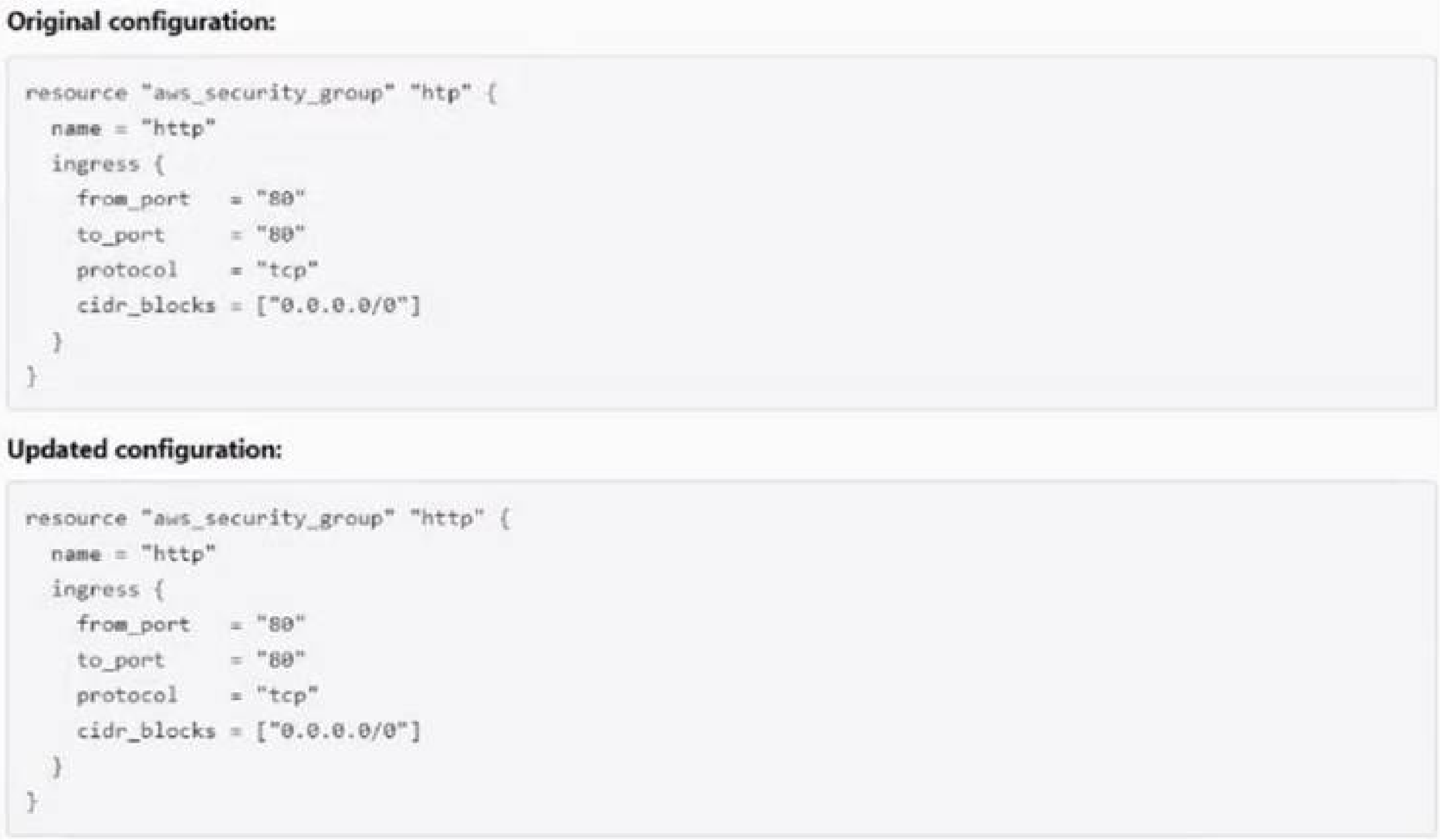
**NEW QUESTION 175**
- (Exam Topic 4)
John is writing a module and within the module, there are multiple places where he has to use the same
conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration,. What is a better approach to dealing with 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 179**
- (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 183**
- (Exam Topic 4)
Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)

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

**Answer:** ABC

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

**NEW QUESTION 187**
- (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 191**
- (Exam Topic 4)
Which parameters does terraform import require? Choose two correct answers.

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

**Answer:** CD

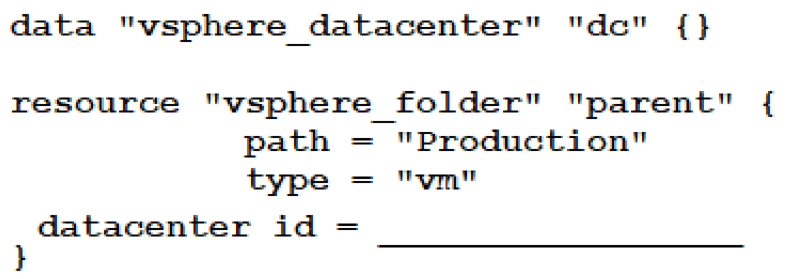
**Explanation:**
https://www.terraform.io/cli/commands/import#usage

**NEW QUESTION 195**
- (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 198**
- (Exam Topic 4)
How would you be able to reference an attribute from the vsphere_datacenter data source for use with the argument within the vsprere_folder resource in the following configuration?



```
data "vsphere_datacenter" "dc" {}

resource "vsphere_folder" "parent" {
            path = "Production"
            type = "vm"
  datacenter id = ____________________
}
```

A. vsphere_datacenter.dc.id
B. data.vsphere_datacenter.dc
C. data.dc,id
D. data.vsphere_datacenter.dc.id

**Answer:** D

**NEW QUESTION 202**
- (Exam Topic 4)
Multiple provider instances blocks for AWS can be part of a single configuration file?

A. False
B. True

**Answer:** B

**Explanation:**
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:
# The default provider configuration provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region provider "aws" {
alias = "west" region = "us-west-2"
}
The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers

that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances

**NEW QUESTION 207**
- (Exam Topic 4)
You have created a main.tr Terraform configuration consisting of an application server, a database, and a load balancer. You ran terraform apply and all resources were created successfully. Now you realize that you do not actually need the load balancer so you run terraform destroy without any flags What will happen?

A. Terraform will destroy the application server because it is listed first in the code
B. Terraform will prompt you to confirm that you want to destroy all the infrastructure
C. Terraform will destroy the main.tf file
D. Terraform will prompt you to pick which resource you want to destroy
E. Terraform will immediately destroy all the infrastructure

**Answer:** B

**NEW QUESTION 208**
- (Exam Topic 4)
What resource dependency information is stored in Terraform's state?

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

**Answer:** B

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

**NEW QUESTION 212**
- (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.

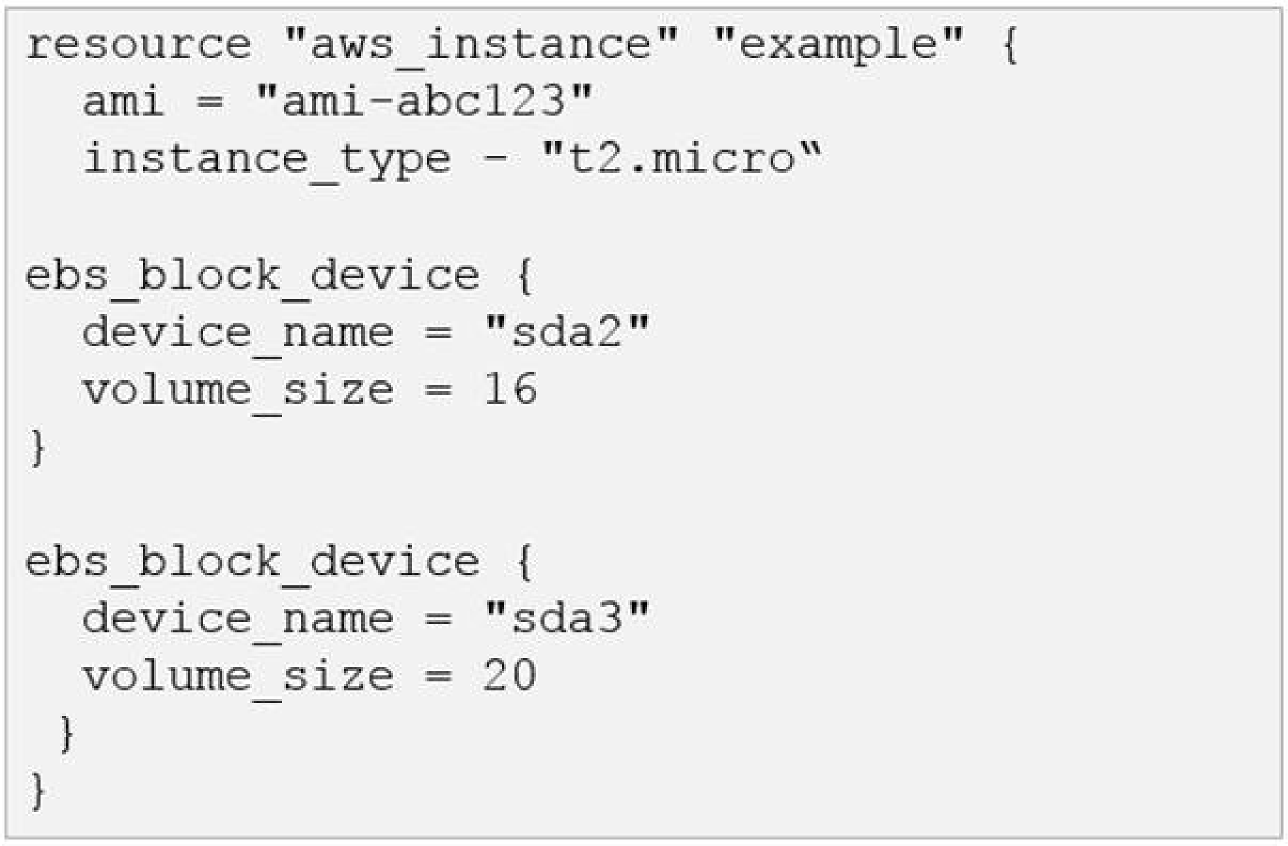
**NEW QUESTION 217**
- (Exam Topic 4)
What is the result of the following terraform function call?

A. hello
B. what?
C. goodbye

**Answer:** B

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

**NEW QUESTION 218**
- (Exam Topic 4)
How would you reference the Volume IDs associated with the ebs_block_device blocks in this configuration?



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

ebs_block_device {
  device_name = "sda2"
  volume_size = 16
}

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

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

**Answer:** A

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

**NEW QUESTION 223**
- (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 226**
- (Exam Topic 4)
Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?

A. False

B. True

**Answer:** B

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

**NEW QUESTION 231**
- (Exam Topic 4)
The following is a snippet from a Terraform configuration file: Which, when validated, results in the following error:
Fill in the blank in the error message with the correct string from the list below.

A. version
B. multi
C. label
D. alias

**Answer:** D

**Explanation:**

https://www.terraform.io/docs/configuration/providers.html#alias-multiple-providerinstances

**NEW QUESTION 232**
- (Exam Topic 4)
Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.
What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

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

**Answer:** D

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

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

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

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

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

**NEW QUESTION 235**
- (Exam Topic 4)
All Terraform Cloud tiers support team management and governance.

A. True
B. False

**Answer:** B

**Explanation:**
https://www.terraform.io/cloud-docs/overview
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. Each higher paid upgrade plan is a strict superset of any lower plans — for example, the Team & Governance plan includes all of the features of the Team plan.

**NEW QUESTION 236**
- (Exam Topic 4)
terraform validate validate validates that your infrastructure matches the Terraform state file.

A. True
B. False

**Answer:** B

**Explanation:**
The terraform validate command validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs, etc. Validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus primarily useful for general verification of reusable modules, including correctness of attribute names and value types. Source: https://www.terraform.io/cli/commands/validate

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

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

**Answer:** B

**NEW QUESTION 241**
- (Exam Topic 4)
Select all Operating Systems that Terraform is available for. (select five)

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

**Answer:** ABDEF

**Explanation:**

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

**NEW QUESTION 244**
- (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 247**
- (Exam Topic 4)
Using multi-cloud and provider-agnostic tools provides which of the following benefits?

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

**Answer:** AC

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

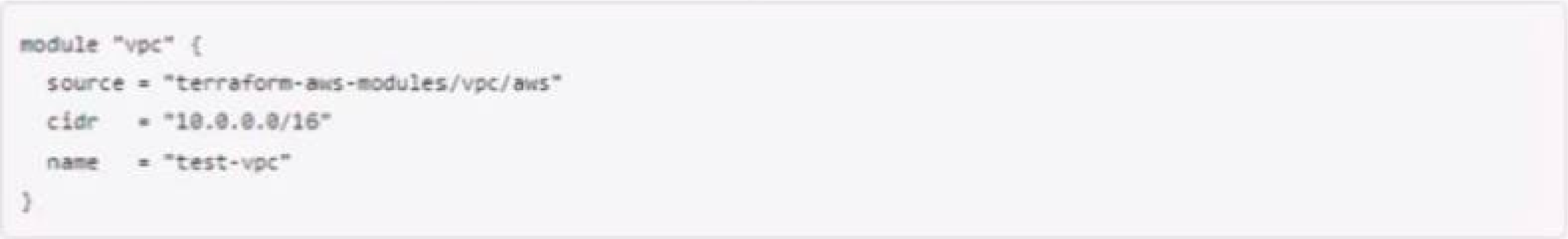
**NEW QUESTION 249**
- (Exam Topic 4)
When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.
For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.

A. Export the value using terraform export and input the value using terraform input.
B. Configure the pertinent provider's configuration with a list of possible IP addresses to use.
C. Configure an output value in the application module in order to use that value for the DNS module.
D. Preconfigure the IP address as a parameter in the DNS module.

**Answer:** C

**Explanation:**
Output values are like the return values of a Terraform module, and have several uses:
* A child module can use outputs to expose a subset of its resource attributes to a parent module.
* A root module can use outputs to print certain values in the CLI output after running terraform apply.
* When using remote state, root module outputs can be accessed by other configurations via a terraform_remote_state data source.
https://www.terraform.io/docs/configuration/outputs.html

**NEW QUESTION 250**
- (Exam Topic 4)
In the below configuration, how would you reference the module output vpc_id ?



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

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

A. Mastered
B. Not Mastered

**Answer:** A

**Explanation:**
module.vpc.id

**NEW QUESTION 253**
- (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 258**
- (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 261**
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

**Answer:** A

**NEW QUESTION 263**
- (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 268**
- (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 269**
- (Exam Topic 4)
Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)

A. Unix
B. Red Hat
C. CentOS
D. Amazon Linux
E. Ubuntu

**Answer:** BCDE

**Explanation:**

https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements

**NEW QUESTION 274**
- (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 279**
- (Exam Topic 4)
What does Terraform use providers for? (Choose three.)

A. Provision resources for on-premises infrastructure services
B. Simplify API interactions
C. Provision resources for public cloud infrastructure services
D. Enforce security and compliance policies
E. Group a collection of Terraform configuration files that map to a single state file

**Answer:** ABC

**NEW QUESTION 284**
- (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 285**
- (Exam Topic 4)
Terraform Enterprise (also referred to as pTFE) requires what type of backend database for a clustered deployment?

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

**Answer:** A

**Explanation:**

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

**NEW QUESTION 290**
- (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

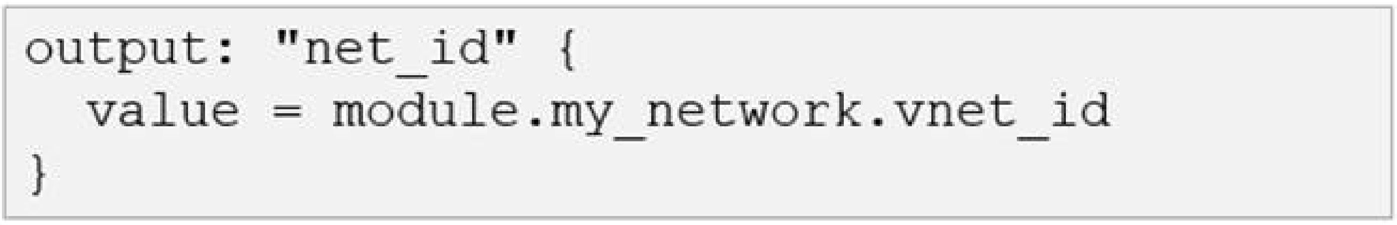
**NEW QUESTION 291**
- (Exam Topic 4)
Which Terraform collection type should you use to store key/value pairs?

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

**Answer:** D

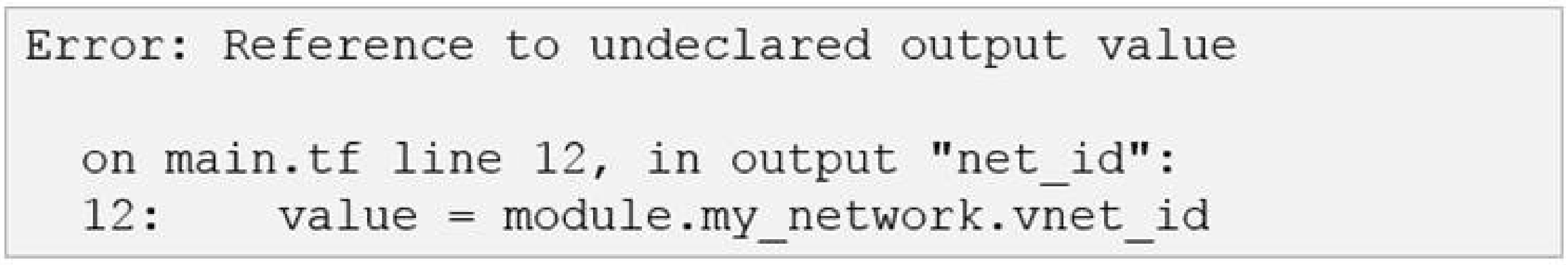
**Explanation:**
Maps/objects are represented by a pair of curly braces containing a series of <KEY> = <VALUE> pairs Source: https://www.terraform.io/language/expressions/types

**NEW QUESTION 294**
- (Exam Topic 4)
You are using a networking module in your Terraform configuration with the name label my_network. In your main configuration you have the following code:



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

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



```
Error: Reference to undeclared output value

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

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

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

**Answer:** C

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

**NEW QUESTION 299**
- (Exam Topic 4)
Which of the following commands will launch the Interactive console for Terraform interpolations?

A. terraform console
B. terraform cli
C. terraform
D. terraform cmdline

**Answer:** B

**Explanation:**
https://www.terraform.io/docs/commands/console.html

**NEW QUESTION 300**

- (Exam Topic 4)
Select all features which are exclusive to Terraform Enterprise. (Select Three)

A. Sentinel
B. Cost Estimation
C. Audit Logs
D. Clustering
E. SAML/SSO

**Answer:** CDE

**Explanation:**
Sentinel and Cost Estimation are also available in Terraform Cloud https://www.hashicorp.com/products/terraform/pricing/

**NEW QUESTION 301**
- (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 305**
- (Exam Topic 4)
Which argument(s) are required when declaring a Terraform variable?

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

**Answer:** E

**Explanation:**
Terraform CLI defines the following OPTIONAL arguments for variable declarations: default - A default value which then makes the variable optional. type - This argument specifies what value types are accepted for the variable. description - This specifies the input variable's documentation. validation - A block to define validation rules, usually in addition to type constraints. sensitive - Limits Terraform UI output when the variable is used in configuration. nullable - Specify if the variable can be null within the module. https://www.terraform.io/language/values/variables#arguments

**NEW QUESTION 310**
- (Exam Topic 4)
In the following code snippet, the block type is identified by which string?

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

**Answer:** B

**NEW QUESTION 315**
......