Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

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

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

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

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

Answer: C

Explanation:

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

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

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

Answer: A

Explanation:

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

NEW QUESTION 3
- (Exam Topic 1)
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)
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 5
- (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 6
- (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 7
- (Exam Topic 1)
Which argument(s) is (are) required when declaring a Terraform variable?

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

Answer: B

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

NEW QUESTION 8
- (Exam Topic 1)
The terraform.tfstate file always matches your currently built infrastructure.
A. True
B. False

Answer: B

Explanation:
Reference: https://www.terraform.io/docs/language/state/index.html

NEW QUESTION 9
- (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 10

- (Exam Topic 1)
Examine the following Terraform configuration, which uses the data source for an AWS AMI. What value should you enter for the ami argument in the AWS instance resource?

```
data "aws_ami" "ubuntu" {
  ...
}

resource "aws_instance" "web" {
  ami = _____________________
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

A. aws_ami.ubuntu
B. data.aws_ami.ubuntu
C. data.aws_ami.ubuntu.id
D. aws_ami.ubuntu.id

Answer: C

Explanation:
resource "aws_instance" "web" { ami= data.aws_ami.ubuntu.id
Reference: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance

NEW QUESTION 10
- (Exam Topic 1)
terraform init initializes a sample main.tf file in the current directory.

A. True
B. False

Answer: B

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

NEW QUESTION 13
- (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 17
- (Exam Topic 1)
Terraform provisioners that require authentication can use the _____ block.

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

Answer: A

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

NEW QUESTION 19
- (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 22
- (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 24
- (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)
What features does the hosted service Terraform Cloud provide? (Choose two.)

A. Automated infrastructure deployment visualization
B. Automatic backups
C. Remote state storage
D. A web-based user interface (UI)

Answer: CD

Explanation:
https://www.terraform.io/enterprise/admin/infrastructure/backup-restore

NEW QUESTION 28
- (Exam Topic 1)
Which provisioner invokes a process on the resource created by Terraform?

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

Answer: A

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." https://www.terraform.io/language/resources/provisioners/local-exec
"The remote-exec provisioner invokes a script on a remote resource after it is created." https://www.terraform.io/language/resources/provisioners/remote-exec

NEW QUESTION 30

- (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 35
- (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 37
- (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 38
- (Exam Topic 1)
How would you reference the "name" value of the second instance of this fictitious resource?

```
resource "aws_instance" "web" {
  count = 2
  name = "terraform-${count.index}"
}
```

A. element(aws_instance.web, 2)
B. aws_instance.web[1].name
C. aws_instance.web[1]
D. aws_instance.web[2].name
E. aws_instance.web.*.name

Answer: B

Explanation:
https://www.terraform.io/language/meta-arguments/count#referring-to-instances Reference: https://www.terraform.io/docs/configuration-0-11/interpolation.html

NEW QUESTION 41
- (Exam Topic 1)
FILL BLANK
Which flag would you add to terraform plan to save the execution plan to a file?
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:
"You can use the optional -out=FILE option to save the generated plan to a file on disk, which you can later execute by passing the file to terraform apply as an extra argument. This two-step workflow is primarily intended for when running Terraform in automation. If you run terraform plan without the -out=FILE option then it will create a speculative plan, which is a description of the effect of the plan but without any intent to actually apply it." https://www.terraform.io/cli/commands/plan

NEW QUESTION 43
- (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 45
- (Exam Topic 1)
When does terraform apply reflect changes in the cloud environment?

A. Immediately
B. However long it takes the resource provider to fulfill the request
C. After updating the state file
D. Based on the value provided to the -refresh command line argument
E. None of the above

Answer: B

NEW QUESTION 50
- (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 54
- (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 55
- (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 56
- (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 58
- (Exam Topic 1)
Which of these is the best practice to protect sensitive values in state files?

A. Blockchain
B. Secure Sockets Layer (SSL)
C. Enhanced remote backends
D. Signed Terraform providers

Answer: C

Explanation:
Use of remote backends and especially the availability of Terraform Cloud, there are now a variety of backends that will encrypt state at rest and will not store the state in cleartext on machines running. Reference:
https://www.terraform.io/docs/extend/best-practices/sensitive-state.html

NEW QUESTION 59
- (Exam Topic 1)
What is the name assigned by Terraform to reference this resource?

```
mainresource “google_compute_instance” “main” {
      name = “test”
}
```

A. compute_instance
B. main
C. google
D. test

Answer: B

NEW QUESTION 62
- (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 64
- (Exam Topic 1)
Which of the following is not a valid Terraform collection type?

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

Answer: C

Explanation:
https://www.terraform.io/language/expressions/type-constraints#collection-types

NEW QUESTION 66
- (Exam Topic 1)
You have deployed a new webapp with a public IP address on a clod provider. However, you did not create any outputs for your code.
What is the best method to quickly find the IP address of the resource you deployed?

A. Run terraform output ip_address to view the result
B. In a new folder, use the terraform_remote_state data source to load in the state file, then write an output for each resource that you find the state file
C. Run terraform state list to find the name of the resource, then terraform state show to find the attributes including public IP address
D. Run terraform destroy then terraform apply and look for the IP address in stdout

Answer: C

Explanation:
https://www.terraform.io/cli/commands/state/show

NEW QUESTION 68
- (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 69
- (Exam Topic 1)
FILL BLANK
You need to specify a dependency manually.
What resource meta-parameter can you use to make sure Terraform respects the dependency?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Reference: https://www.terraform.io/docs/language/functions/file.html

NEW QUESTION 70
- (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 72
- (Exam Topic 1)
All standard backend types support state storage, locking, and remote operations like plan. apply and destroy.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/language/settings/backends/configuration
"Some of these backends act like plain remote disks for state files, while others support locking the state while operations are being performed. This helps prevent conflicts and inconsistencies. The built-in backends listed are the only backends. You cannot load additional backends as plugins."

NEW QUESTION 74
- (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 75
- (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 76
- (Exam Topic 1)
Which backend does the Terraform CLI use by default?

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

Answer: D

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

NEW QUESTION 77
- (Exam Topic 2)
When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

A. False
B. True

Answer: B

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

NEW QUESTION 79
- (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 83
- (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 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 89
- (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 92
- (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 97
- (Exam Topic 2)
Provisioners should only be used as a last resort.

A. False
B. True

Answer: B

Explanation:
Provisioners are a Last Resort
Terraform includes the concept of provisioners as a measure of pragmatism, knowing that there will always be certain behaviors that can't be directly represented in Terraform's declarative model.
However, they also add a considerable amount of complexity and uncertainty to Terraform usage. Firstly, Terraform cannot model the actions of provisioners as part of a plan because they can in principle take any action. Secondly, successful use of provisioners requires coordinating many more details than Terraform usage usually requires: direct network access to your servers, issuing Terraform credentials to log in, making sure that all of the necessary external software is installed, etc.
The following sections describe some situations which can be solved with provisioners in principle, but where better solutions are also available. We do not recommend using provisioners for any of the use-cases described in the following sections.
Even if your specific use-case is not described in the following sections, we still recommend attempting to solve it using other techniques first, and use provisioners only if there is no other option.
https://www.terraform.io/docs/provisioners/index.html

NEW QUESTION 98
- (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 99
- (Exam Topic 2)
You want to use terraform import to start managing infrastructure that was not originally provisioned through infrastructure as code. Before you can import the resource's current state, what must you do in order to prepare to manage these resources using Terraform?

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

Answer: B

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

NEW QUESTION 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 108
- (Exam Topic 2)
terraform state subcommands such as list are read-only commands, do read-only commands create state backup files?

A. Yes
B. No

Answer: B

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

NEW QUESTION 110
- (Exam Topic 2)
lookup retrieves the value of a single element from which of the below data type?

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

Answer: A

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

NEW QUESTION 112
- (Exam Topic 2)
Terraform import command can import resources into modules as well directly into the root of your state.

A. True
B. False

Answer: A

Explanation:
Import will find the existing resource from ID and import it into your Terraform state at the given ADDRESS. ADDRESS must be a valid resource address. Because any resource address is valid, the import command can import resources into modules as well directly into the root of your state.
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. For 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.
As a result of the above command, the resource is recorded in the state file. We can now run terraform plan to see how the configuration compares to the imported resource, and make any adjustments to the configuration to align with the current (or desired) state of the imported object.
https://www.terraform.io/docs/commands/import.html

NEW QUESTION 113
- (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 117
- (Exam Topic 2)
terraform refresh command will not modify infrastructure, but does modify 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.
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION 119
- (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 122
- (Exam Topic 2)
You have declared a variable name my_var in terraform configuration without a value associated with it. variable my_var {}
After running terraform plan it will show an error as variable is not defined.

A. True
B. False

Answer: B

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

"2xCPU-4GB" = "80"
}
}
These can then be used to find the right storage size based on the monthly price as defined in the previous example.
size = lookup(var.storage_sizes, var.plans["5USD"])
Boolean
The last of the available variable type is boolean. They give the option to employ simple true or false values. For example, you might wish to have a variable that decides when to generate the root user password on a new deployment.
variable "set_password" { default = false
}
The above example boolean can be used similarly to a string variable by simply marking down the correct variable.
create_password = var.set_password
By default, the value is set to false in this example. However, you can overwrite the variable at deployment by assigning a different value in a command-line variable.
terraform apply -var set_password="true"

NEW QUESTION 127
- (Exam Topic 2)
You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?

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

Answer: C

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

NEW QUESTION 129
- (Exam Topic 2)
What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

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

Answer: D

Explanation:
Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION 131
- (Exam Topic 2)
By default, a defined provisioner is a creation-time provisioner.

A. True
B. False

Answer: A

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

NEW QUESTION 134
- (Exam Topic 2)
Refer to the below code where developer is outputting the value of the database password but has used sensitive parameter to hide the output value in the CLI.
output "db_password" { value = aws_db_instance.db.password description = "The password for logging in to the database." sensitive = true}
Since sensitive is set to true, the value associated with db password will not be present in state file as plain-text?

A. False
B. True

Answer: A

Explanation:
Sensitive output values are still recorded in the state, and so will be visible to anyone who is able to access the state data.

NEW QUESTION 136
- (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 138
- (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 142
- (Exam Topic 2)
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a _____.

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

Answer: D

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

NEW QUESTION 146
- (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 147
- (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 149
- (Exam Topic 2)
What does terrafom plan do ?

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

Answer: A

NEW QUESTION 150
- (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 154
- (Exam Topic 3)
You are reviewing Terraform configurations for a big project in your company. You noticed that there are several identical sets of resources that appear in multiple configurations. What feature of Terraform would you recommend to use to reduce the amount of cloned configuration between the different configurations?

A. Packages
B. Backends
C. Provisioners
D. Modules

Answer: D

Explanation:
Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.
A module is a container for multiple resources that are used together. Modules can be used to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.
Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.
https://www.terraform.io/docs/modules/index.html

NEW QUESTION 156
- (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 157
- (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 158
- (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 163
- (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 166
- (Exam Topic 3)
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. What command will do this?

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

Answer: A

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

NEW QUESTION 167
- (Exam Topic 3)
Which of the following value will be accepted for var1? variable "var1" {
type = string
}

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

Answer: B

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

NEW QUESTION 169
- (Exam Topic 3)
Which of the following allows Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code?

A. Sentinel
B. Module registry
C. Functions
D. Workspaces

Answer: A

Explanation:
Sentinel is a language and framework for policy built to be embedded in existing software to enable
fine-grained, logic-based policy decisions. A policy describes under what circumstances certain behaviors are allowed. Sentinel is an enterprise-only feature.
https://www.youtube.com/watch?v=Vy8s7AAvU6g&feature=emb_title

NEW QUESTION 173
- (Exam Topic 3)
State locking does not happen automatically and must be specified at run

A. False
B. True

Answer: A

Explanation:
State locking happens automatically on all operations that could write state. https://www.terraform.io/docs/state/locking.html

NEW QUESTION 177
- (Exam Topic 3)
* 1. resource "aws_s3_bucket" "example" {
* 2. bucket = "my-test-s3-terraform-bucket"
* 3. …} resource "aws_iam_role" "test_role" {
* 4. name = "test_role"
* 5. …}
Due to the way that the application code is written, the s3 bucket must be created before the test role is created, otherwise there will be a problem. How can you ensure that?

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

Answer: A

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

NEW QUESTION 179
- (Exam Topic 3)
Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.

A. Set TF_LOG = DEBUG
B. Set VAR_TF = TRACE
C. Set TF_LOG = TRACE
D. Set VAR_TF_LOG = TRACE

Answer: C

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

NEW QUESTION 180
- (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 184
- (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 189
- (Exam Topic 3)
Which flag would be used within a Terraform configuration block to identify the specific version of a provider required?

A. required-provider
B. required-version
C. required_providers
D. required_versions

Answer: C

Explanation:
For production use, you should constrain the acceptable provider versions via configuration file to ensure that new versions with breaking changes will not be automatically installed by terraform init in the future.
Example terraform {
required_providers { aws = ">= 2.7.0"
}
}

NEW QUESTION 194
- (Exam Topic 3)
When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?

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

Answer: B

Explanation:
If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state.
State locking happens automatically on all operations that could write state. You won't see any message that it is happening. If state locking fails, Terraform will not continue. You can disable state locking for most commands with the -lock flag but it is not recommended.
If acquiring the lock is taking longer than expected, Terraform will output a status message. If Terraform doesn't output a message, state locking is still occurring if your backend supports it.
Not all backends support locking. Please view the list of backend types for details on whether a backend supports locking or not.
https://www.terraform.io/docs/state/locking.html

NEW QUESTION 197
- (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 201
- (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 204
- (Exam Topic 3)
Terraform Enterprise currently supports running under which the following operating systems?

A. Ubuntu
B. Amazon Linux
C. Debian
D. CentOS
E. Red Hat Enterprise Linux
F. Oracle Linux

Answer: ABCDEF

Explanation:
Terraform Enterprise runs on Linux instances, and you must prepare a running Linux instance for Terraform Enterprise before running the installer. You will start and manage this instance like any other server.
Terraform Enterprise currently supports running under the following operating systems: Standalone deployment:
Debian 7.7+
Ubuntu 14.04.5 / 16.04 / 18.04
Red Hat Enterprise Linux 7.4 - 7.8 CentOS 6.x / 7.4 - 7.8
Amazon Linux 2014.03 / 2014.09 / 2015.03 / 2015.09 / 2016.03 / 2016.09 / 2017.03 / 2017.09 / 2018.03 / 2.0
Oracle Linux 7.4 - 7.8 https://www.terraform.io/docs/enterprise/before-installing/index.html

NEW QUESTION 206
- (Exam Topic 3)
You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.

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

Answer: B

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

NEW QUESTION 211
- (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 212
- (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 216
- (Exam Topic 3)
What kind of resource dependency is stored in terraform.tfstate file?

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

Answer: A

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 218
- (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 223
- (Exam Topic 3)
Hanah is writing a terraform configuration with nested modules, there are multiple places where she has to use the same conditional expression but she wants to avoid repeating the same values or expressions multiple times in the configuration,. What is a better approach to dealing with this?

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

Answer: B

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

NEW QUESTION 225
- (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 230
- (Exam Topic 3)
Which of the following are string functions? Select three

A. tostring
B. tonumber
C. Chomp
D. format

E. join

Answer: CDE

Explanation:
tonumber and tostring are Type Conversion function https://www.terraform.io/docs/configuration/functions.html

NEW QUESTION 233
- (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 234
- (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 235
- (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 237
- (Exam Topic 3)
You have provisioned some aws resources in your test environment through Terraform for a POC work. After the POC, now you want to destroy the resources but before destroying them you want to check what resources will be getting destroyed through terraform. what are the options of doing that? (Select TWO)

A. Use terraform destroy command
B. This is not possible
C. Use terraform plan command
D. Use terraform plan -destroy command.

Answer: AD

Explanation:
https://learn.hashicorp.com/terraform/getting-started/destroy

NEW QUESTION 240
- (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 242
- (Exam Topic 4)
terraform init retrieves the source code tot all referenced modules

A. True
B. False

Answer: A

Explanation:
Terraform installs providers, initialises source code & modules etc at this stage

NEW QUESTION 243
- (Exam Topic 4)
How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? (Choose two.)
A. A full audit trail of the request and fulfillment process is generated
B. A request must be submitted for infrastructure changes
C. As additional resources are required, more tickets are submitted
D. A catalog of approved resources can be accessed from drop down lists in a request form

Answer: BC

NEW QUESTION 245
- (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 246
- (Exam Topic 4)
Select the answer below that completes the following statement: Terraform Cloud can be managed from the CLI but requires _____?

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

Answer: A

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

NEW QUESTION 250
- (Exam Topic 4)
HashiCorp offers multiple versions of Terraform, including Terraform open-source, Terraform Cloud, and Terraform Enterprise. Which of the following Terraform features are only available in the Enterprise edition? (select four)

A. SAML/SSO
B. Sentinel
C. Audit Logs
D. Clustering
E. Private Module Registry
F. Private Network Connectivity

Answer: ACF

Explanation:

While there are a ton of features that are available to open source users, many features that are part of the Enterprise offering are geared towards larger teams and enterprise functionality. To see what specific features are part of Terraform Cloud and Terraform Enterprise, check out this link.
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 255
- (Exam Topic 4)
What is the purpose of a Terraform workspace in either open source of enterprise?

A. Workspaces allow you to manage collections of infrastructure in state files.
B. A logical separation of business units
C. A method of grouping multiple infrastructure security policies
D. Provides limited access to a cloud environment

Answer: B

NEW QUESTION 257
- (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 260
- (Exam Topic 4)
Terraform is currently being used by your organisation to create resources on AWS for the development of a web application. One of your coworkers wants to change the instance type to "t2.large" while keeping the default set values.
What adjustments does the teammate make in order to meet his goal?

A. Issue Terraform plan instance.type".t2.large" and it deploys the instance
B. Modify the tf.variableswith the instance type and issue terraform apply
C. Create a new file my.tfvars and add the type of the instance and issue terraform plan and apply
D. Modify the terraform.tfvars with the instance type and issue terraform plan and then terraform apply to deploy the instances

Answer: D

NEW QUESTION 265
- (Exam Topic 4)
In the example below, the depends_on argument creates what type of dependency?

A. implicit dependency
B. internal dependency
C. explicit dependency
D. non-dependency resource

Answer: C

NEW QUESTION 267
- (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 271
- (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 276
- (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 280
- (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 281
- (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 284
- (Exam Topic 4)
What are the benefits of using Infrastructure as Code? (select five)

A. Infrastructure as Code is relatively simple to learn and write, regardless of a user's prior experience with developing code
B. Infrastructure as Code provides configuration consistency and standardization among deployments
C. Infrastructure as Code is easily repeatable, allowing the user to reuse code to deploy similar, yet different resources
D. Infrastructure as Code gives the user the ability to recreate an application's infrastructure for disaster recovery scenarios

E. Infrastructure as Code easily replaces development languages such as Go and .Net for application development
F. Infrastructure as Code allows a user to turn a manual task into a simple, automated deployment (Correct)

Answer: ACDF

Explanation:

If you are new to infrastructure as code as a concept, it is the process of managing infrastructure in a file or files rather than manually configuring resources in a user interface.
A resource in this instance is any piece of infrastructure in a given environment, such as a virtual machine, security group, network interface, etc. At a high level, Terraform allows operators to use HCL to author files containing definitions of their desired resources on almost any provider (AWS, GCP, GitHub, Docker, etc) and automates the creation of those resources at the time of application.

NEW QUESTION 288
- (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 293
- (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 297
- (Exam Topic 4)
Why should secrets not be hard coded into Terraform code? Choose two correct answers

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

Answer: BC

NEW QUESTION 298
- (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 302
- (Exam Topic 4)
Which of the following statements best describes the Terraform list(...) type?

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

Answer: B

Explanation:
A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.
https://www.terraform.io/docs/configuration/types.html#structural-types

NEW QUESTION 306
- (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 310
- (Exam Topic 4)
What does this code do?

```
terraform {
  required_providers {
    aws = "~> 3.0"
  }
}
```

A. Requires any version of the AWS provider > = 3.0 and < 4.0
B. Requires any version of the AWS provider > = 3.0
C. Requires any version of the AWS provider after the 3.0 major release like 4.1
D. Requires any version of the AWS provider > 3.0

Answer: A

Explanation:
https://www.terraform.io/language/expressions/version-constraints#-3
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

NEW QUESTION 313
- (Exam Topic 4)
In the example below, where is the value of the DNS record's IP address originating from?
* 1. resource "aws_route53_record" "www"
* 2. {
* 3. zone_id = aws_route53_zone.primary.zone_id
* 4. name = "www.example.com"
* 5. type = "A"
* 6. ttl = "300"
* 7. records = [module.web_server.instance_ip_address] 8. }

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

Answer: B

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

NEW QUESTION 318
- (Exam Topic 4)
Terraform variable names are saved in the state file.

A. True
B. False

Answer: B

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

NEW QUESTION 320
- (Exam Topic 4)
Which of the following is true about Terraform's implementation of infrastructure as code? (Choose two.)

A. It is only compatible with AWS infrastructure management
B. You cannot reuse infrastructure configuration
C. You can version your infrastructure configuration
D. It requires manual configuration of infrastructure resources
E. It allows you to automate infrastructure provisioning

Answer: CE

NEW QUESTION 322
- (Exam Topic 4)
From the code below, identify the implicit dependency:

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

Answer: C

NEW QUESTION 324
- (Exam Topic 4)
Which statements best describes what the local variable assignment is doing in the following code snippet:

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

Answer: D

NEW QUESTION 329
- (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 330

- (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 334
- (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 339
- (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 344
- (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 346
- (Exam Topic 4)
terraform destroy is the only way to remove infrastructure.

A. True
B. False

Answer: B

NEW QUESTION 348
- (Exam Topic 4)
Resources in terraform can have same identifiers(Resource type + Block name).

A. True
B. False

Answer: B

NEW QUESTION 349
- (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 353
- (Exam Topic 4)
You decide to move a Terraform state file to Amazon S3 from another location. You write the code below into a file called\

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

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

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

Answer: B

NEW QUESTION 355
- (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 359
- (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 361
- (Exam Topic 4)
What feature of Terraform Cloud and/or Terraform Enterprise can you publish and maintain a set of custom modules which can be used within your organization?

A. Terraform registry
B. custom VCS integration
C. private module registry
D. remote runs

Answer: C

NEW QUESTION 366
- (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 368
- (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 371
- (Exam Topic 4)
A terraform apply can not _____ infrastructure.

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

Answer: A

NEW QUESTION 373
- (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 377
- (Exam Topic 4)
You have modified your local Terraform configuration and ran terraform plan to review the changes. Simultaneously, your teammate manually modified the infrastructure component you are working on. Since you already ran terraform plan locally, the execution plan for terraform apply will be the same.

A. True
B. False

Answer: B

NEW QUESTION 378
- (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 382
- (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 386
- (Exam Topic 4)
In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to use what?

A. Input Parameters
B. Module
C. Regular Expressions
D. Output Value

Answer: A

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

NEW QUESTION 390
- (Exam Topic 4)
Which of the following is not supported backend types in Terra form?

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

Answer: D

NEW QUESTION 392
- (Exam Topic 4)
You have written a terraform IaC script which was working till yesterday , but is giving some vague error from today , which you are unable to understand . You want more detailed logs that could potentially help you troubleshoot the issue , and understand the root cause. What can you do to enable this setting? Please note , you are using terraform OSS.

A. Terraform OSS can push all its logs to a syslog endpoin
B. As such, you have to set up the syslog sink, and enable TF_LOG_PATH env variable to the syslog endpoint and all logs will automatically start streaming.
C. Detailed logs are not available in terraform OSS, except the crash messag
D. You need to upgrade to terraform enterprise for this point.
E. Enable the TF_LOG_PATH to the log sink file location, and logging output will automatically be stored there.
F. Enable TF_LOG to the log level DEBUG, and then set TF_LOG_PATH to the log sink file location.Terraform debug logs will be dumped to the sink path, even in terraform OSS.

Answer: D

Explanation:
Terraform has detailed logs which 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.
To persist logged output you can set TF_LOG_PATH in order to force the log to always be appended to a specific file when logging is enabled. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

NEW QUESTION 395
- (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 399
- (Exam Topic 4)
From the answers below, select the advantages of using Infrastructure as Code.

A. Provide a codified workflow to develop customer-facing applications.
B. Safely test modifications using a "dry run" before applying any actual changes.
C. Easily integrate with application workflows (GitLab Actions, Azure DevOps, CI/CD tools).
D. Easily change and update existing infrastructure.
E. Provide reusable modules for easy sharing and collaboration.

Answer: BCDE

Explanation:
Infrastructure as Code is not used to develop applications, but it can be used to help deploy or provision those applications to a public cloud provider or on-premises infrastructure.
All of the others are benefits to using Infrastructure as Code over the traditional way of managing infrastructure, regardless if it's public cloud or on-premises.

NEW QUESTION 404
- (Exam Topic 4)
Which of the following is a meta-argument defined in the configuration files of Terraform?

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

Answer: B

NEW QUESTION 409
- (Exam Topic 4)
Choose the answer that correctly completes the sentence: _____ backends support state locking.

A. All
B. No
C. Only local
D. Some

Answer: D

NEW QUESTION 413
- (Exam Topic 4)
Named workspaces are not a suitable isolation mechanism for strong separation between staging and production?

A. True
B. False

Answer: A

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

NEW QUESTION 418
- (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 419
- (Exam Topic 4)
If a DevOps team adopts AWS Cloud Formation as their standardized method for provisioning public cloud resources, which of the following scenarios poses a challenge for this team?

A. The team is asked to manage a new application stack built on AWS-natrve services
B. The organization decides to expand into Azure and wishes to deploy new infrastructure using their existing codebase
C. The team is asked to build a reusable code base that can deploy resources into any AWS region
D. The DevOps team is tasked with automating a manual provisioning process

Answer: B

NEW QUESTION 423
- (Exam Topic 4)
You've used Terraform to deploy a virtual machine and a database. You want to replace this virtual machine instance with an identical one without affecting the database. What is the best way to achieve this using Terraform?

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

Answer: A

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

NEW QUESTION 427
- (Exam Topic 4)
You cannot install third party plugins using terraform init.

A. True
B. False

Answer: B

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

NEW QUESTION 430
- (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 431
- (Exam Topic 4)
Terraform plan updates your state file.

A. True
B. False

Answer: B

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

NEW QUESTION 435
- (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 440
- (Exam Topic 4)
Suppose terraformcode is taking up some values which are not defined inside the code files. In which of the following options issue might have occurred?

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

Answer: D

NEW QUESTION 444
- (Exam Topic 4)
How do you specify a module's version when publishing it to the public Terraform Module Registry?

A. The module's configuration page on the Terraform Module Registry
B. Terraform Module Registry does not support versioning modules
C. The release tags in the associated repo Most Voted
D. The module's Terraform code

Answer: C

Explanation:
https://www.terraform.io/registry/modules/publish

NEW QUESTION 447
- (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 450
- (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 455
- (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 459
- (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 463
- (Exam Topic 4)
Once a new Terraform backend is configured with a Terraform code block, which command(s) is (are) used to migrate the state file?

A. terraform apply
B. terraform push
C. terraform destroy, then terraform apply
D. terraform init

Answer: B

Explanation:
https://www.terraform.io/cli/commands/state/push

NEW QUESTION 468
- (Exam Topic 4)
When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)

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

Answer: BCD

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

NEW QUESTION 472
- (Exam Topic 4)
You need to write some Terraform code that adds 42 firewall rules to a security group as shown in the example.

```
resource "aws_security_group" "many_rules" {
  name = "many-rules"
  ingress {
    from_port = 443
    to_port = 443
    protocol = "tcp"
    cidr_blocks = "0.0.0.0/0"
  }
}
```

What can you use to avoid writing 42 different nested ingress config blocks by hand?

A. A count loop
B. A for block
C. A for each block
D. A 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. Reference: https://www.terraform.io/language/expressions/dynamic-blocks

NEW QUESTION 474
.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual TA-002-P Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the TA-002-P Product From:

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

Money Back Guarantee

TA-002-P Practice Exam Features:

* TA-002-P Questions and Answers Updated Frequently

* TA-002-P Practice Questions Verified by Expert Senior Certified Staff

* TA-002-P Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* TA-002-P Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year