HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

NEW QUESTION 1
- (Exam Topic 1)
What features stops multiple admins from changing the Terraform state at the same time?

A. Version control
B. Backend types
C. Provider constraints
D. State locking

Answer: D

Explanation:
Somewhat ambiguous question however the key phrase is "feature". You need a remote backend first with a State Locking feature available to avoid this scenario.
https://blog.gruntwork.io/how-to-manage-terraform-state-28f5697e68fa

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

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

Answer: A

Explanation:

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

NEW QUESTION 3
- (Exam Topic 1)
You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you delete the VM using the cloud provider console, and run terraform apply again without changing any Terraform code?

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

Answer: D

NEW QUESTION 4
- (Exam Topic 1)
Terraform can only manage resource dependencies if you set them explicitly with the depends_on argument.

A. True
B. False

Answer: A

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

NEW QUESTION 5
- (Exam Topic 1)
How can terraform plan aid in the development process?

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

Answer: A

Explanation:
"The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. By default, when Terraform creates a plan it:
Reads the current state of any already-existing remote objects to make sure that the Terraform state is up-to-date.
Compares the current configuration to the prior state and noting any differences.
Proposes a set of change actions that should, if applied, make the remote objects match the configuration."
"The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review.
If Terraform detects that no changes are needed to resource instances or to root module output values, terraform plan will report that no actions need to be taken."

https://www.terraform.io/cli/commands/plan

NEW QUESTION 6
- (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 7
- (Exam Topic 1)
You're building a CI/CD (continuous integration/ continuous delivery) pipeline and need to inject sensitive variables into your Terraform run.
How can you do this safely?

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

Answer: A

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

NEW QUESTION 8
- (Exam Topic 1)
You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (files). You need to enable debug messages to find this out.
Which of the following would achieve this?

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

Answer: A

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

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

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

Answer: B

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

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

A. True
B. False

Answer: B

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

NEW QUESTION 11
- (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 14
- (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 18
- (Exam Topic 1)
Module variable assignments are inherited from the parent module and do not need to be explicitly set.

A. True
B. False

Answer: B

NEW QUESTION 20
- (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 24
- (Exam Topic 1)
You need to constrain the GitHub provider to version 2.1 or greater.
Which of the following should you put into the Terraform 0.12 configuration's provider block?

A. version >= 2.1
B. version ~> 2.1
C. version = "<= 2.1"
D. version = ">= 2.1"

Answer: D

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

NEW QUESTION 29
- (Exam Topic 1)
You have never used Terraform before and would like to test it out using a shared team account for a cloud provider. The shared team account already contains 15 virtual machines (VM). You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully.
What should you do to delete the newly-created VM with Terraform?

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

Answer: B

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

NEW QUESTION 33
- (Exam Topic 1)
What is the workflow for deploying new infrastructure with Terraform?

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

Answer: D

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

NEW QUESTION 37
- (Exam Topic 1)
If a module declares a variable with a default, that variable must also be defined within the module.

A. True
B. False

Answer: B

NEW QUESTION 40
- (Exam Topic 1)
Terraform variables and outputs that set the "description" argument will store that description in the state file.

A. True
B. False

Answer: B

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

NEW QUESTION 41
- (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 45
- (Exam Topic 1)
What is not processed when running a terraform refresh?

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

Answer: B

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

NEW QUESTION 48
- (Exam Topic 1)
A Terraform local value can reference other Terraform local values.

A. True
B. False

Answer: A

Explanation:
"The expressions in local values are not limited to literal constants; they can also reference other values in the module in order to transform or combine them, including variables, resource attributes, or other local values:" https://www.terraform.io/language/values/locals#declaring-a-local-value

NEW QUESTION 51
- (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 54
- (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 58
- (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 60
- (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 64
- (Exam Topic 1)
What command does Terraform require the first time you run it within a configuration directory?

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

Answer: B

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

NEW QUESTION 66
- (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 68
- (Exam Topic 1)
How is terraform import run?

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

Answer: D

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

NEW QUESTION 69
- (Exam Topic 2)
Which one of the following command will rewrite Terraform configuration files to a canonical format and style.

A. terraform graph -h
B. terraform init
C. terraform graph
D. terraform fmt

Answer: D

Explanation:
The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style. This command applies a subset of the Terrra Terraform language style conventions, along with other minor adjustments for readability.

NEW QUESTION 70
- (Exam Topic 2)
Environment variables can be used to set variables. The environment variables must be in the format " "_<variablename>. Select the correct prefix string from the following list.

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

Answer: C

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

NEW QUESTION 72
- (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 75
- (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 79
- (Exam Topic 2)
Terraform has detailed logs which can be enabled by setting the _____ environmental variable.

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

Answer: C

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

NEW QUESTION 81
- (Exam Topic 2)
You want terraform plan and apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?

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

Answer: C

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

NEW QUESTION 83
- (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 86
- (Exam Topic 2)
How can you ensure that the engineering team who has access to git repo will not create any non-compliant resources that might lead to a security audit failure in future. your team is using Hashicorp Terraform Enterprise Edition.

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

Answer: C

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

NEW QUESTION 91
- (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 93
- (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 95
- (Exam Topic 2)
Terraform works well in Windows but a Windows server is required.

A. False
B. True

Answer: A

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

NEW QUESTION 97
- (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 98
- (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 102
- (Exam Topic 2)
Which of the below configuration file formats are supported by Terraform? (Select TWO)

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

Answer: BE

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

NEW QUESTION 104
- (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 107
- (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 110
- (Exam Topic 3)
In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

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

Answer: D

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

NEW QUESTION 111
- (Exam Topic 3)

Which of the following is the right substitute for static values that can make Terraform configuration file more dynamic and reusable?

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

Answer: B

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

NEW QUESTION 112
- (Exam Topic 3)
Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.

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

Answer: B

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

NEW QUESTION 113
- (Exam Topic 3)
Which of the below features of Terraform can be used for managing small differences between different environments which can act more like completely separate working directories.

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

Answer: B

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

NEW QUESTION 118
- (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 122
- (Exam Topic 3)

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

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

Answer: D

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

NEW QUESTION 125
- (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 128
- (Exam Topic 3)
After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?

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

Answer: D

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

NEW QUESTION 130
- (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 131
- (Exam Topic 3)
In regards to Terraform state file, select all the statements below which are correct?

A. When using local state, the state file is stored in plain-text.

B. The state file is always encrypted at rest.
C. Storing state remotely can provide better security.
D. Using the mask feature, you can instruct Terraform to mask sensitive data in the state file.
E. The Terraform state can contain sensitive data, therefore the state file should be protected from unauthorized access.
F. Terraform Cloud always encrypts state at rest.

Answer: ACEF

Explanation:
Terraform state can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.
When using local state, state is stored in plain-text JSON files.
When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.
Storing Terraform state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.
Recommendations
If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.
Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.
For example:
* 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.
* The S3 backend supports encryption at rest when the encrypt option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

NEW QUESTION 135
- (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 139
- (Exam Topic 3)
What happens when a terraform apply command is executed?

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

Answer: B

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

NEW QUESTION 140
- (Exam Topic 3)
You want terraform plan and terraform apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?

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

Answer: D

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

NEW QUESTION 144
- (Exam Topic 3)
Which of the below command will upgrade the provider version to the latest acceptable one?

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

Answer: C

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

NEW QUESTION 147
- (Exam Topic 3)
Which of the below datatype is not supported by Terraform.

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

Answer: A

NEW QUESTION 151
- (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 152
- (Exam Topic 3)
Terraform-specific settings and behaviors are declared in which configuration block type?

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

Answer: B

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

NEW QUESTION 153
- (Exam Topic 3)
By default, provisioners that fail will also cause the Terraform apply itself to error. How can you change this default behavior within a provisioner?

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

Answer: C

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

NEW QUESTION 154
- (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 157
- (Exam Topic 3)
You have created a terraform script that uses a lot of new constructs that have been introduced in terraform v0.12. However, many developers who are cloning the script from your git repo, are using v0.11, and getting errors. What can be done from your end to solve this problem?

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

Answer: A

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

NEW QUESTION 161
- (Exam Topic 3)
Mary has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code:
* 1. output "db_password"
* 2. {
* 3. value = local.db_password
* 4. }
Mary wants to hide the output value in the CLI after terraform apply? What is the best way?

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

Answer: B

NEW QUESTION 163
- (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 164
- (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 168
- (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 172
- (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 174
- (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 177
- (Exam Topic 4)
What is the result of the following terraform function call?

A. True
B. False

Answer: B

Explanation:

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

NEW QUESTION 181
- (Exam Topic 4)
A user has created a module called "my_test_module" and committed it to GitHub. Over time, several commits have been made with updates to the module, each tagged in GitHub with an incremental version number. Which of the following lines would be required in a module configuration block in terraform to select tagged version v1.0.4?

A. source = "git::https://example.com/my_test_module.git@tag=v1.0.4"
B. source = "git::https://example.com/my_test_module.git&ref=v1.0.4"
C. source = "git::https://example.com/my_test_module.git#tag=v1.0.4"
D. source = "git::https://example.com/my_test_module.git?ref=v1.0.4"

Answer: D

Explanation:
https://www.terraform.io/docs/modules/sources.html#selecting-a-revision

NEW QUESTION 183
- (Exam Topic 4)
What does the command terraform fmt do?

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

Answer: A

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

https://www.terraform.io/docs/commands/fmt.html

NEW QUESTION 186
- (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 188
- (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 190
- (Exam Topic 4)
Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 …) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region. What should you do? Your provider setting is as below
The default provider configuration provider "aws" { region = "us-east-1" }

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

Answer: C

Explanation:
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:
# The default provider configuration provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region provider "aws" {
alias = "west" region = "us-west-2"
}
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION 193
- (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 194
- (Exam Topic 4)
Which of the following value will be accepted for my_var?
* 1. variable "my_var"
* 2. {
* 3. type = string
* 4. }

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

Answer: C

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

NEW QUESTION 198
- (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 199
- (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 201
- (Exam Topic 4)
Which of the following is not a benefit of adopting infrastructure as code?

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

Answer: D

NEW QUESTION 206
- (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 207
- (Exam Topic 4)
Which parameters does terraform import require? Choose two correct answers.

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

Answer: CD

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

NEW QUESTION 211
- (Exam Topic 4)
Which task does teraform init not perform?

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

Answer: B

NEW QUESTION 212
- (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 214
- (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 219
- (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 220
- (Exam Topic 4)
What resource dependency information is stored in Terraform's state?

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

Answer: B

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

NEW QUESTION 222
- (Exam Topic 4)
Resources in terraform can have same identifiers(Resource type + Block name).

A. True
B. False

Answer: B

NEW QUESTION 226
- (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 228
- (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 231
- (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 234
- (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 237
- (Exam Topic 4)
In a Terraform Cloud workspace linked to a version control repository, speculative plan runs start automatically when you merge or commit changes to version control.

A. True
B. False

Answer: B

NEW QUESTION 238
- (Exam Topic 4)
Module version is required to reference a module on the Terraform Module Registry.

A. True
B. False

Answer: B

NEW QUESTION 241
- (Exam Topic 4)
You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed?

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

Answer: A

Explanation:
Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.
To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.

NEW QUESTION 246
- (Exam Topic 4)
A terraform apply can not _____ infrastructure.

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

Answer: A

NEW QUESTION 249
- (Exam Topic 4)
Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

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

Answer: B

Explanation:

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

NEW QUESTION 254
- (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 256
- (Exam Topic 4)
You can reference a resource created with for_each using a Splat (*) expression.

A. True
B. False

Answer: B

Explanation:
Splat Expressions with Maps The splat expression patterns shown above apply only to lists, sets, and tuples. To get a similar result with a map or object value you

must use for expressions. Resources that use the for_each argument will appear in expressions as a map of objects, so you can't use splat expressions with those resources. For more information, see Referring to Resource Instances. https://www.terraform.io/language/meta-arguments/for_each#referring-to-instances https://www.terraform.io/language/expressions/references

NEW QUESTION 257
- (Exam Topic 4)
Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.
What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

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

Answer: D

Explanation:
Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.
This is a great way to slowly transition infrastructure to Terraform.
The terraform import command is used to import existing infrastructure.
To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.
Example:
resource "aws_instance" "import_example" {
# ...instance configuration...
}
Now terraform import can be run to attach an existing instance to this resource configuration.
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside
Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 260
- (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 264
- (Exam Topic 4)
Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)

A. GitHub
B. CVS Version Control
C. Azure DevOps Server
D. Bitbucket Cloud
E. GitHub Enterprise

Answer: ACDE

Explanation:

Terraform Cloud supports the following VCS providers:
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github-enterprise.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-com.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html

- https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers

NEW QUESTION 266
- (Exam Topic 4)
While Terraform is generally written using the HashiCorp Configuration Language (HCL), what other syntax can Terraform are expressed in?

A. JSON
B. YAML
C. TypeScript
D. XML

Answer: A

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

NEW QUESTION 268
- (Exam Topic 4)
Why does this backend configuration not follow best practices?

```
terraform {
  backend "s3" {
    bucket     = "terraform-state-prod"
    key        = "network/terraform.tfstate"
    region     = "us-east-1"
    access_key = "AKIAIOSFODNN7EXAMPLE"
    secret_key = "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
  }

  required_providers {
      aws = {
        source  = "hashicorp/aws"
        version = "~> 3.38"
      }
    }

    required_version = ">= 0.15"
}
```

A. You should not store credentials in Terraform Configuration
B. You should use the local enhanced storage backend whenever possible
C. An alias meta-argument should be included in backend blocks whenever possible
D. The backend configuration should contain multiple credentials so that more than one user can execute terraform plan and terraform apply

Answer: A

NEW QUESTION 269
- (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 273
- (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 277

- (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 280
- (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 281
- (Exam Topic 4)
You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you terraform apply again immediately afterwards without changing any Terraform code?

A. Terraform will terminate and recreate the VM
B. Terraform will create another duplicate VM
C. Terraform will apply the VM to the state file
D. Nothing

Answer: D

NEW QUESTION 285
- (Exam Topic 4)
You want to share Terraform state with your team, store it securely and provide state locking. How would you do this? Choose three correct answers.

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

Answer: ABE

NEW QUESTION 288
- (Exam Topic 4)
resource "aws_s3_bucket" "example" { bucket = "my-test-s3-terraform-bucket" ...} resource "aws_iam_role" "test_role" { name = "test_role" ...}
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. This will already be taken care of by terraform native implicit dependenc
B. Nothing else needs to be done from your end.
C. Add explicit dependency using depends_on . This will ensure the correct order of resource creation.
D. 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.
E. 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.

Answer: B

Explanation:
Use the depends_on meta-argument to handle hidden resource dependencies that Terraform can't automatically infer.
Explicitly specifying a dependency is only necessary when a resource relies on some other resource's behavior but doesn't access any of that resource's data in its arguments.

NEW QUESTION 293
- (Exam Topic 4)
In the below configuration, how would you reference the module output vpc_id ?

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

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

A. Mastered
B. Not Mastered

Answer: A

Explanation:
module.vpc.id

NEW QUESTION 294
- (Exam Topic 4)
You need to specify a dependency manually. What resource meta-parameter can you use lo make sure Terraform respects thee dependency?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
depends_on

NEW QUESTION 297
- (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 300
- (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 305
- (Exam Topic 4)
Which of the following is the safest way to inject sensitive values into a Terraform Cloud workspace?

A. Write the value to a file and specify the file with the -var-file flag
B. Set a value for the variable in the UI and check the "Sensitive" check box
C. Edit the state file directly just before running terraform apply
D. Set the variable value on the command line with the -var flag

Answer: B

Explanation:
-var and -var-file overwrite workspace-specific and variable set variables that have the same key. From the workspace, variable can be added and checked off as being sensitive. Reference: https://www.terraform.io/cloud-docs/workspaces/variables/managing-variables#loading-variables-from-files
https://www.terraform.io/cloud-docs/workspaces/variables

NEW QUESTION 309
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

Answer: A

NEW QUESTION 314
- (Exam Topic 4)
Which of the following does terraform apply change after you approve the execution plan? Choose two correct answers.

A. The execution plan
B. Terraform code

C. Cloud infrastructure
D. State file
E. The .terraform directory

Answer: CD

NEW QUESTION 318
- (Exam Topic 4)
What does state locking accomplish?

A. Copies the state file from memory to disk
B. Encrypts any credentials stored within the state file
C. Blocks Terraform commands from modifying the state file
D. Prevents accidental deletion of the state file

Answer: C

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. Source: https://www.terraform.io/language/state/locking

NEW QUESTION 322
- (Exam Topic 4)
A junior admin accidentally deleted some of your cloud instances. What does Terraform do when you run terraform apply?

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

Answer: C

NEW QUESTION 324
- (Exam Topic 4)
Which is the best way to specify a tag of v1.0.0 when referencing a module stored in Git (for example git::https://example.com/vpc.git)?

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

Answer: A

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

NEW QUESTION 329
- (Exam Topic 4)
Your team lead does not trust the junior terraform engineers who now have access to the git repo . So , he wants you to have some sort of a checking layer , whereby , you can ensure that the juniors will not create any non-compliant resources that might lead to a security audit failure in future. What can you do to efficiently enforce this?

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

Answer: B

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

NEW QUESTION 334
- (Exam Topic 4)
Your firm employs a version control system (for example, git) and has requested that you commit all terraform code to it. During the commit, you must be cautious with sensitive information. Which of the following files should be left out of the commit?

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

Answer:

D

NEW QUESTION 339
- (Exam Topic 4)
You have just developed a new Terraform configuration for two virtual machines with a cloud provider. You would like to create the infrastructure for the first time. Which Terraform command should you run first?

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

Answer: D

NEW QUESTION 344
- (Exam Topic 4)
What Terraform feature is shown in the example below?

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

Answer: C

NEW QUESTION 347
- (Exam Topic 4)
Which of the following is not a valid Terraform string function?

A. replace
B. format
C. join
D. tostring

Answer: D

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

NEW QUESTION 352
- (Exam Topic 4)
When writing Terraform code, HashiCorp recommends that you use how many spaces between each nesting level?

A. 1
B. 2
C. 4

Answer: C

Explanation:
The Terraform parser allows you some flexibility in how you lay out the elements in your configuration files, but the Terraform language also has some idiomatic style conventions which we recommend users always follow for consistency between files and modules written by different teams. Automatic source code formatting tools may apply these conventions automatically.
Indent two spaces for each nesting level.
When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs:
ami = "abc123" instance_type = "t2.micro"
When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them.
Use one blank line to separate the arguments from the blocks.
Use empty lines to separate logical groups of arguments within a block.
For blocks that contain both arguments and "meta-arguments" (as defined by the Terraform language semantics), list meta-arguments first and separate them from other arguments with one blank line. Place meta-argument blocks last and separate them from other blocks with one blank line.
resource "aws_instance" "example" { count = 2 # meta-argument first
ami = "abc123" instance_type = "t2.micro" network_interface {
# ...
}
lifecycle { # meta-argument block last create_before_destroy = true
}
}
Top-level blocks should always be separated from one another by one blank line. Nested blocks should also be separated by blank lines, except when grouping together related blocks of the same type (like multiple provisioner blocks in a resource).
Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example: root_block_device, ebs_block_device and
ephemeral_block_device on aws_instance form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

NEW QUESTION 355
- (Exam Topic 4)
Given the Terraform configuration below, in which order will the resources be created?

A. Larger image
B. resources will be created simultaneously
C. aws_eip will be created first aws_instance will be created second
D. aws_instance will be created first aws_eip will be created second

Answer: D

Explanation:
The aws_instance will be created first, and then aws_eip will be created second due to the aws_eip's resource dependency of the aws_instance id

NEW QUESTION 359
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

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