HashiCorp.Terraform Associate 003.by.Usy.73q

Number: Terraform Associate 003
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: Terraform Associate 003

Exam Name: HashiCorp Certified: Terraform Associate

Exam A

QUESTION 1
What is one disadvantage of using dynamic blocks in Terraform?

A. Dynamic blocks can construct repeatable nested blocks
B. Terraform will run more slowly
C. They cannot be used to loop through a list of values
D. They make configuration harder to read and understand

Correct Answer: D
Section:
Explanation:
This is one disadvantage of using dynamic blocks in Terraform, as they can introduce complexity and reduce readability of the configuration. The other options are either advantages or incorrect statements.

QUESTION 2
Which backend does the Terraform CU use by default?

A. Depends on the cloud provider configured
B. HTTP
C. Remote
D. Terraform Cloud
E. Local

Correct Answer: E
Section:
Explanation:
This is the backend that the Terraform CLI uses by default, unless you specify a different backend in your configuration. The local backend stores the state file in a local file namedterraform.tfstate, which can be used to track and manage the state of your infrastructure.

QUESTION 3
How does Terraform manage most dependencies between resources?

A. Terraform will automatically manage most resource dependencies
B. Using the depends_on parameter
C. By defining dependencies as modules and including them in a particular order
D. The order that resources appear in Terraform configuration indicates dependencies

Correct Answer: A
Section:
Explanation:
This is how Terraform manages most dependencies between resources, by using the references between them in the configuration files. For example, if resource A depends on resource B, Terraform will create resource B first and then pass its attributes to resource A.

QUESTION 4

You should run terraform fnt to rewrite all Terraform configurations within the current working directory to conform to Terraform-style conventions.

A. True
B. False

Correct Answer: A
Section:
Explanation:
You should runterraform fmtto rewrite all Terraform configurations within the current working directory to conform to Terraform-style conventions. This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability. It is recommended to use this command to ensure consistency of style across different Terraform codebases. The command is optional, opinionated, and has no customization options, but it can help you and your team understand the code more quickly and easily.Reference= :Command: fmt:Using Terraform fmt Command to Format Your Terraform Code

QUESTION 5
Your risk management organization requires that new AWS S3 buckets must be private and encrypted at rest. How can Terraform Cloud automatically and proactively enforce this security control?

A. Auditing cloud storage buckets with a vulnerability scanning tool
B. By adding variables to each Terraform Cloud workspace to ensure these settings are always enabled
C. With an S3 module with proper settings for buckets
D. With a Sentinel policy, which runs before every apply

Correct Answer: D
Section:
Explanation:
The best way to automatically and proactively enforce the security control that new AWS S3 buckets must be private and encrypted at rest is with a Sentinel policy, which runs before every apply. Sentinel is a policy as code framework that allows you to define and enforce logic-based policies for your infrastructure. Terraform Cloud supports Sentinel policies for all paid tiers, and can run them before anyterraform planorterraform applyoperation. You can write a Sentinel policy that checks the configuration of the S3 buckets and ensures that they have the proper settings for privacy and encryption, and then assign the policy to your Terraform Cloud organization or workspace. This way, Terraform Cloud will prevent any changes that violate the policy from being applied.Reference= [Sentinel Policy Framework], [Manage Policies in Terraform Cloud], [Write and Test Sentinel Policies for Terraform]

QUESTION 6
Which of the following is not a key principle of infrastructure as code?

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

Correct Answer: D
Section:
Explanation:
The key principle of infrastructure as code that is not listed among the options isgolden images. Golden images are pre-configured, ready-to-use virtual machine images that contain a specific set of software and configuration. They are often used to create multiple identical instances of the same environment, such as for testing or production. However, golden images are not a principle of infrastructure as code, but rather a technique that can be used with or without infrastructure as code. The other options are all key principles of infrastructure as code, as explained below:
Self-describing infrastructure: This means that the infrastructure is defined in code that describes its desired state, rather than in scripts that describe the steps to create it. This makes the infrastructure easier to understand, maintain, and reproduce.
Idempotence: This means that applying the same infrastructure code multiple times will always result in the same state, regardless of the initial state. This makes the infrastructure consistent and predictable, and avoids errors or conflicts caused by repeated actions.

QUESTION 7
A developer on your team is going lo leaf down an existing deployment managed by Terraform and deploy a new one. However, there is a server resource named aws instant.ubuntu[l] they would like to keep. What command should they use to tell Terraform to stop managing that specific resource?

A. Terraform plan rm:aws_instance.ubuntu[1]
B. Terraform state rm:aws_instance.ubuntu[1]
C. Terraform apply rm:aws_instance.ubuntu[1]
D. Terraform destory rm:aws_instance.ubuntu[1]

Correct Answer: B
Section:
Explanation:
To tell Terraform to stop managing a specific resource without destroying it, you can use theterraform state rmcommand. This command will remove the resource from the Terraform state, which means that Terraform will no longer track or update the corresponding remote object. However, the object will still exist in the remote system and you can later useterraform importto start managing it again in a different configuration or workspace. The syntax for this command isterraform state rm , whereis the resource address that identifies the resource instance to remove. For example,terraform state rm aws_instance.ubuntu[1]will remove the second instance of theaws_instanceresource namedubuntufrom the state.Reference= :Command: state rm:Moving Resources

QUESTION 8
HashiCorp Configuration Language (HCL) supports user-denned functions.

A. True
B. False

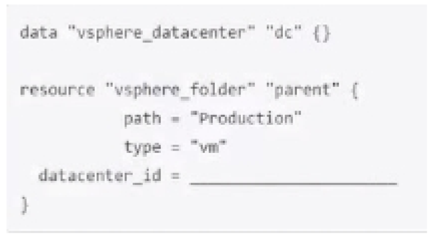
Correct Answer: B
Section:
Explanation:
HashiCorp Configuration Language (HCL) does not support user-defined functions. You can only use the built-in functions that are provided by the language. The built-in functions allow you to perform various operations and transformations on values within expressions. The general syntax for function calls is a function name followed by comma-separated arguments in parentheses, such asmax(5, 12, 9). You can find the documentation for all of the available built-in functions in the Terraform Registry or the Packer Documentation, depending on which tool you are using.Reference= :Functions - Configuration Language | Terraform:Functions - Configuration Language | Packer

QUESTION 9
Terraform providers are part of the Terraform core binary.

A. True
B. False

Correct Answer: B
Section:
Explanation:
Terraform providers are not part of the Terraform core binary. Providers are distributed separately from Terraform itself and have their own release cadence and version numbers. Providers are plugins that Terraform uses to interact with various APIs, such as cloud providers, SaaS providers, and other services. You can find and install providers from the Terraform Registry, which hosts providers for most major infrastructure platforms. You can also load providers from a local mirror or cache, or develop your own custom providers. To use a provider in your Terraform configuration, you need to declare it in the provider requirements block and optionally configure its settings in the provider block.Reference= :Providers - Configuration Language | Terraform:Terraform Registry - Providers Overview | Terraform

QUESTION 10
How could you reference an attribute from the vsphere_datacenter data source for use with the datacenter_id argument within the vsphere_folder resource in the following configuration?



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

resource "vsphere_folder" "parent" {
           path = "Production"
           type = "vm"
  datacenter_id = ______________________
}
```

A. Data.vsphere_datacenter.DC.id
B. Vsphere_datacenter.dc.id
C. Data,dc,id
D. Data.vsphere_datacenter,dc

Correct Answer: A
Section:
Explanation:
The correct way to reference an attribute from the vsphere_datacenter data source for use with the datacenter_id argument within the vsphere_folder resource in the following configuration isdata.vsphere_datacenter.dc.id. This follows the syntax for accessing data source attributes, which isdata.TYPE.NAME.ATTRIBUTE. In this case, the data source type isvsphere_datacenter, the data source name isdc, and the attribute we want to access isid. The other options are incorrect because they either use the wrong syntax, the wrong punctuation, or the wrong case.Reference= [Data Source: vsphere_datacenter], [Data Source: vsphere_folder], [Expressions: Data Source Reference]

QUESTION 11
Which of the following module source paths does not specify a remote module?

A. Source = ''module/consul''
B. Source = ''githhub.comicrop/example''
C. Source =''git@github.com:hasicrop/example.git''
D. Source = ''hasicrop/consul/aws''

Correct Answer: A
Section:
Explanation:
The module source path that does not specify a remote module issource = 'module/consul'. This specifies a local module, which is a module that is stored in a subdirectory of the current working directory. The other options are all examples of remote modules, which are modules that are stored outside of the current working directory and can be accessed by various protocols, such as Git, HTTP, or the Terraform Registry. Remote modules are useful for sharing and reusing code across different configurations and environments.Reference= [Module Sources], [Local Paths], [Terraform Registry], [Generic Git Repository], [GitHub]

QUESTION 12
Terraform configuration (including any module references) can contain only one Terraform provider type.

A. True
B. False

Correct Answer: B
Section:
Explanation:
Terraform configuration (including any module references) can contain more than one Terraform provider type. Terraform providers are plugins that Terraform uses to interact with various cloud services and other APIs. A Terraform configuration can use multiple providers to manage resources across different platforms and services. For example, a configuration can use the AWS provider to create a virtual machine, the Cloudflare provider to manage DNS records, and the GitHub provider to create a repository. Terraform supports hundreds of providers for different use cases and scenarios.Reference= [Providers], [Provider Requirements], [Provider Configuration]

QUESTION 13
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 correct answers.

A. [var.list [*] , id]
B. [for o in var.list : o.Id]
C. var.list[*].id
D. { for o in var.llst : o => o.id }

Correct Answer: B, C
Section:
Explanation:
These are two ways to produce a list of the IDs from a list of objects that have an attributeid, using either a for expression or a splat expression syntax.

QUESTION 14
What does the default 'local' Terraform backend store?

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

Correct Answer: B
Section:
Explanation:
The default ''local'' Terraform backend stores the state file in a local file named terraform.tfstate, which can be used to track and manage the state of your infrastructure3.

QUESTION 15
Which of these commands makes your code more human readable?

A. Terraform validate
B. Terraform output
C. Terraform show
D. Terraform fmt

Correct Answer: D
Section:
Explanation:
The command that makes your code more human readable isterraform fmt. This command is used to rewrite Terraform configuration files to a canonical format and style, following the Terraform language style conventions and other minor adjustments for readability. The command is optional, opinionated, and has no customization options, but it is recommended to ensure consistency of style across different Terraform codebases. Consistency can help your team understand the code more quickly and easily, making the use ofterraform fmtvery important. You can run this command on your configuration files before committing them to source control or as part of your CI/CD pipeline.Reference= :Command: fmt:Using Terraform fmt Command to Format Your Terraform Code

QUESTION 16
Any user can publish modules to the public Terraform Module Registry.

A. True
B. False

Correct Answer: A
Section:
Explanation:
The Terraform Registry allows any user to publish and share modules. Published modules support versioning, automatically generate documentation, allow browsing version histories, show examples and READMEs, and more. Public modules are managed via Git and GitHub, and publishing a module takes only a few minutes.Once a module is published, releasing a new version of a module is as simple as pushing a properly formed Git tag1.
Reference= The information can be verified from the Terraform Registry documentation on Publishing Modules provided by HashiCorp Developer1.

QUESTION 17
Which task does terraform init not perform?

A. Validates all required variables are present
B. Sources any modules and copies the configuration locally
C. Connects to the backend
D. Sources all providers used in the configuration and downloads them

Correct Answer: A
Section:
Explanation:
The terraform init command is used to initialize a working directory containing Terraform configuration files. This command performs several different initialization steps to prepare the current working directory for use with Terraform, which includes initializing the backend, installing provider plugins, and copying any modules referenced in the configuration. However, it does not validate whether all required variables are present; that is a task performed by terraform plan or terraform apply1.
Reference = This information can be verified from the official Terraform documentation on the terraform init command provided by HashiCorp Developer1.

QUESTION 18
Which of the following is not a benefit of adopting infrastructure as code?

A. Versioning
B. A Graphical User Interface
C. Reusability of code
D. Automation

Correct Answer: B
Section:
Explanation:
Infrastructure as Code (IaC) provides several benefits, including the ability to version control infrastructure, reuse code, and automate infrastructure management. However, IaC is typically associated with declarative configuration files and does not inherently provide a graphical user interface (GUI). A GUI is a feature that may be provided by specific tools or platforms built on top of IaC principles but is not a direct benefit of IaC itself1.
Reference = The benefits of IaC can be verified from the official HashiCorp documentation on ''What is Infrastructure as Code with Terraform?'' provided by HashiCorp Developer1.

QUESTION 19
Which of these is true about Terraform's plugin-based architecture?

A. Terraform can only source providers from the internet
B. Every provider in a configuration has its own state file for its resources
C. You can create a provider for your API if none exists
D. All providers are part of the Terraform core binary

Correct Answer: C
Section:

Explanation:
Terraform is built on a plugin-based architecture, enabling developers to extend Terraform by writing new plugins or compiling modified versions of existing plugins1. Terraform plugins are executable binaries written in Go that expose an implementation for a specific service, such as a cloud resource, SaaS platform, or API2. If there is no existing provider for your API, you can create one using the Terraform Plugin SDK3 or the Terraform Plugin Framework4. Reference =
* 1: Plugin Development - How Terraform Works With Plugins | Terraform | HashiCorp Developer
* 2: Lab: Terraform Plug-in Based Architecture - GitHub
* 3: Terraform Plugin SDK - Terraform by HashiCorp
* 4: HashiCorp Terraform Plugin Framework Now Generally Available

QUESTION 20
The Terraform binary version and provider versions must match each other in a single configuration.

A. True
B. False

Correct Answer: B
Section:
Explanation:
The Terraform binary version and provider versions do not have to match each other in a single configuration. Terraform allows you to specify provider version constraints in the configuration's terraform block, which can be different from the Terraform binary version1. Terraform will use the newest version of the provider that meets the configuration's version constraints2. You can also use the dependency lock file to ensure Terraform is using the correct provider version3. Reference =
* 1: Providers - Configuration Language | Terraform | HashiCorp Developer
* 2: Multiple provider versions with Terraform - Stack Overflow
* 3: Lock and upgrade provider versions | Terraform - HashiCorp Developer

QUESTION 21
Which of the following is not true of Terraform providers?

A. An individual person can write a Terraform Provider
B. A community of users can maintain a provider
C. HashiCorp maintains some providers
D. Cloud providers and infrastructure vendors can write, maintain, or collaborate on Terraform
E. providers
F. None of the above

Correct Answer: F
Section:
Explanation:
All of the statements are true of Terraform providers. Terraform providers are plugins that enable Terraform to interact with various APIs and services1. Anyone can write a Terraform provider, either as an individual or as part of a community2. HashiCorp maintains some providers, such as the AWS, Azure, and Google Cloud providers3. Cloud providers and infrastructure vendors can also write, maintain, or collaborate on Terraform providers, such as the VMware, Oracle, and Alibaba Cloud providers. Reference =
* 1: Providers - Configuration Language | Terraform | HashiCorp Developer
* 2: Plugin Development - How Terraform Works With Plugins | Terraform | HashiCorp Developer
* 3: Terraform Registry
* : Terraform Registry

QUESTION 22
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 Terraform Cloud organization owners can approve plans in VCS connected workspaces
D. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces

Correct Answer: B
Section:
Explanation:
This will trigger a run in the Terraform Cloud workspace, which will perform a plan and apply operation on the infrastructure defined by the Terraform configuration files in the VCS repository.

QUESTION 23
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. Copy the sensitive variables into your Terraform code
B. Store the sensitive variables in a secure_varS.tf file
C. Store the sensitive variables as plain text in a source code repository
D. Pass variables to Terraform with a -var flag

Correct Answer: D
Section:
Explanation:
This is a secure way to inject sensitive variables into your Terraform run, as they will not be stored in any file or source code repository. You can also use environment variables or variable files with encryption to pass sensitive variables to Terraform.

QUESTION 24
When should you write Terraform configuration files for existing infrastructure that you want to start managing with Terraform?

A. You can import infrastructure without corresponding Terraform code
B. Terraform will generate the corresponding configuration files for you
C. Before you run terraform Import
D. After you run terraform import

Correct Answer: C
Section:
Explanation:
You need to write Terraform configuration files for the existing infrastructure that you want to import into Terraform, otherwise Terraform will not know how to manage it. The configuration files should match the type and name of the resources that you want to import.

QUESTION 25
Variables declared within a module are accessible outside of the module.

A. True
B. False

Correct Answer: B
Section:
Explanation:
Variables declared within a module are only accessible within that module, unless they are explicitly exposed as output values1.

QUESTION 26
Your security team scanned some Terraform workspaces and found secrets stored in plaintext in state files. How can you protect that data?

A. Edit your state file to scrub out the sensitive data
B. Always store your secrets in a secrets.tfvars file
C. Delete the state file every time you run Terraform
D. Store the state in an encrypted backend

Correct Answer: D
Section:
Explanation:
This is a secure way to protect sensitive data in the state file, as it will be encrypted at rest and in transit2. The other options are not recommended, as they could lead to data loss, errors, or security breaches.

QUESTION 27
If you update the version constraint in your Terraform configuration, Terraform will update your lock file the next time you run terraform Init.

A. True
B. False

Correct Answer: A
Section:
Explanation:
If you update the version constraint in your Terraform configuration, Terraform will update your lock file the next time you runterraform init3. This will ensure that you use the same provider versions across different machines and runs.

QUESTION 28
Once you configure a new Terraform backend with a terraform code block, which command(s) should you use to migrate the state file?

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

Correct Answer: A
Section:
Explanation:
This command will initialize the new backend and prompt you to migrate the existing state file to the new location4. The other commands are not relevant for this task.

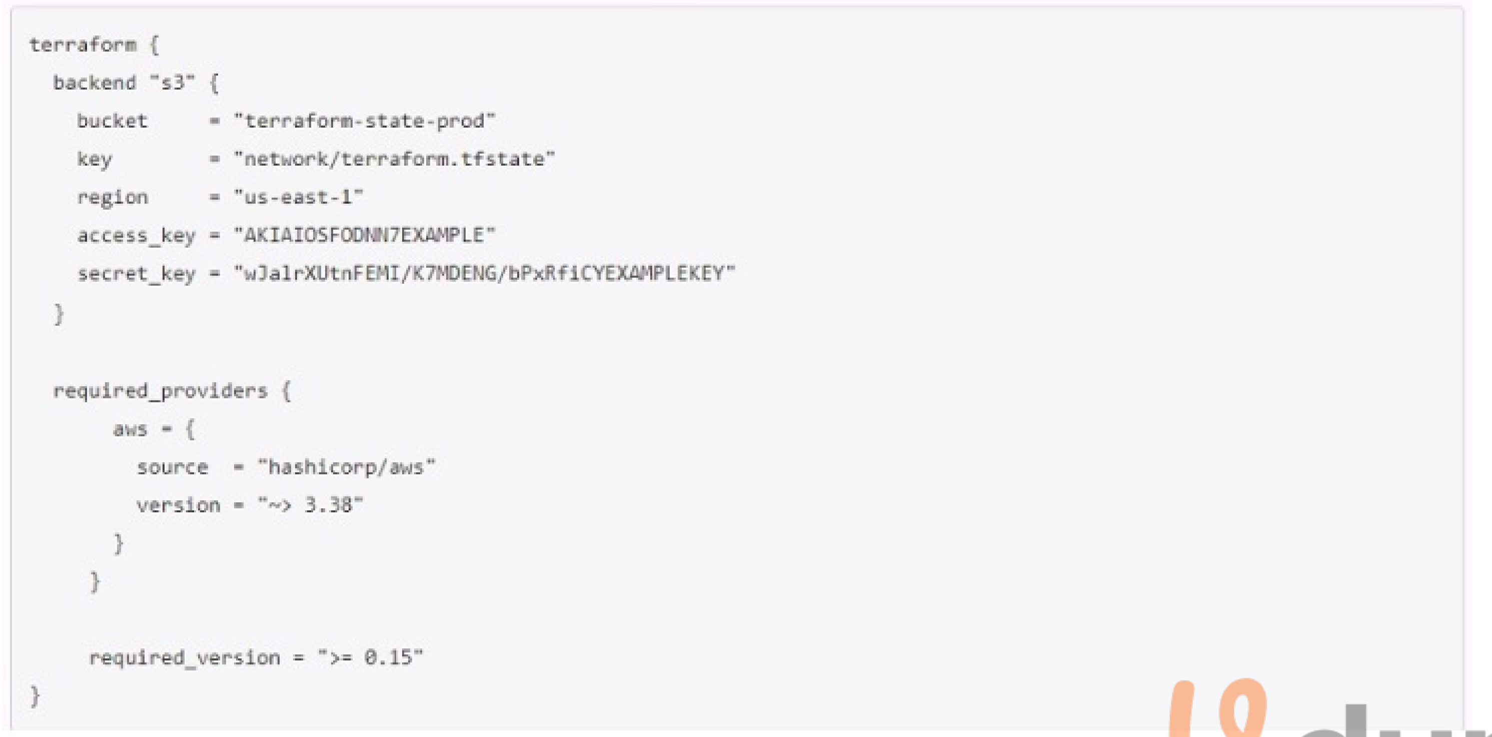
QUESTION 29
What does Terraform use the .terraform.lock.hc1 file for?

A. There is no such file
B. Tracking specific provider dependencies
C. Preventing Terraform runs from occurring
D. Storing references to workspaces which are locked

Correct Answer: B
Section:

Explanation:
The.terraform.lock.hclfile is a new feature in Terraform 0.14 that records the exact versions of each provider used in your configuration. This helps ensure consistent and reproducible behavior across different machines and runs.

QUESTION 30
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. An alias meta-argument should be included in backend blocks whenever possible
B. You should use the local enhanced storage backend whenever possible
C. You should not store credentials in Terraform configuration
D. The backend configuration should contain multiple credentials so that more than one user can execute terraform plan and terraform apply

Correct Answer: C
Section:
Explanation:
This is a bad practice, as it exposes your credentials to anyone who can access your configuration files or state files. You should use environment variables, credential files, or other mechanisms to provide credentials to Terraform.

QUESTION 31
It is best practice to store secret data in the same version control repository as your Terraform configuration.

A. True
B. False

Correct Answer: B
Section:
Explanation:
It is not a best practice to store secret data in the same version control repository as your Terraform configuration, as it could expose your sensitive information to unauthorized parties or compromise your security. You should use environment variables, vaults, or other mechanisms to store and provide secret data to Terraform.

QUESTION 32
________backends support state locking.

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

Correct Answer: C
Section:
Explanation:
Some backends support state locking, which prevents other users from modifying the state file while a Terraform operation is in progress. This prevents conflicts and data loss. Not all backends support this feature, and you can check the documentation for each backend type to see if it does.

QUESTION 33
You want to define a single input variable to capture configuration values for a server. The values must represent memory as a number, and the server name as a string.
Which variable type could you use for this input?

A. List
B. Object
C. Map
D. Terraform does not support complex input variables of different types

Correct Answer: B
Section:
Explanation:
This is the variable type that you could use for this input, as it can store multiple attributes of different types within a single value. The other options are either invalid or incorrect for this use case.

QUESTION 34
Which of the following statements about Terraform modules is not true?

A. Modules can call other modules
B. A module is a container for one or more resources
C. Modules must be publicly accessible
D. You can call the same module multiple times

Correct Answer: C
Section:
Explanation:
This is not true, as modules can be either public or private, depending on your needs and preferences. You can use the Terraform Registry to publish and consume public modules, or use Terraform Cloud or Terraform Enterprise to host and manage private modules.

QUESTION 35
When you use a remote backend that needs authentication, HashiCorp recommends that you:

A. Write the authentication credentials in the Terraform configuration files
B. Keep the Terraform configuration files in a secret store

C. Push your Terraform configuration to an encrypted git repository
D. Use partial configuration to load the authentication credentials outside of the Terraform code

Correct Answer: D
Section:
Explanation:
This is the recommended way to use a remote backend that needs authentication, as it allows you to provide the credentials via environment variables, command-line arguments, or interactive prompts, without storing them in the Terraform configuration files.

QUESTION 36
You can reference a resource created with for_each using a Splat (*) expression.

A. True
B. False

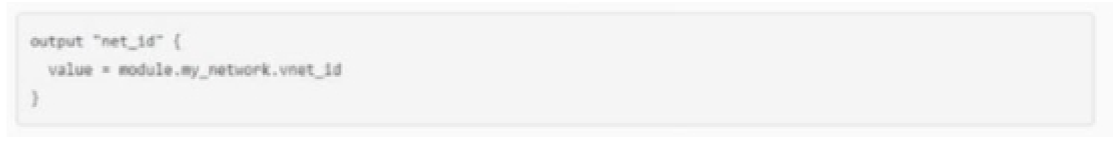
Correct Answer: B
Section:
Explanation:
You cannot reference a resource created withfor_eachusing a splat (*) expression, as it will not work with resources that have non-numeric keys. You need to use aforexpression instead to iterate over the resource instances.

QUESTION 37
What type of block is used to construct a collection of nested configuration blocks?

A. Dynamic
B. For_each
C. Nesting
D. repeated.

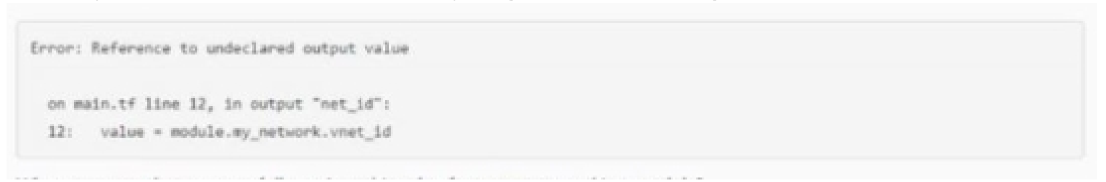
Correct Answer: A
Section:
Explanation:
This is the type of block that is used to construct a collection of nested configuration blocks, by using afor_eachargument to iterate over a collection value and generate a nested block for each element. For example, you can use a dynamic block to create multiple ingress rules for a security group resource.

QUESTION 38
You are using a networking module in your Terraform configuration with the name label my-network. In your main configuration you have the following code:



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

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



```
Error: Reference to undeclared output value

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

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

A. Change the reference value to my-network,outputs,vmet_id

B. Define the attribute vmet_id as a variable in the networking modeule
C. Define the attribute vnet_id as an output in the networking module
D. Change the reference value module.my,network,outputs,vnet_id

Correct Answer: C
Section:
Explanation:
This is what you must do to successfully retrieve this value from your networking module, as it will expose the attribute as an output value that can be referenced by other modules or resources. The error message indicates that the networking module does not have an output value namedvnet_id, which causes the reference to fail.

QUESTION 39
Where does the Terraform local backend store its state?

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

Correct Answer: C
Section:
Explanation:
This is where the Terraform local backend stores its state, by default, unless you specify a different file name or location in your configuration. The local backend is the simplest backend type that stores the state file on your local disk.

QUESTION 40
As a developer, you want to ensure your plugins are up to date with the latest versions. Which Terraform command should you use?

A. terraform refresh -upgrade
B. terraform apply -upgrade
C. terraform init -upgrade
D. terraform providers -upgrade

Correct Answer: C
Section:
Explanation:
This command will upgrade the plugins to the latest acceptable version within the version constraints specified in the configuration. The other commands do not have an-upgradeoption.

QUESTION 41
What information does the public Terraform Module Registry automatically expose about published modules?

A. Required input variables
B. Optional inputs variables and default values
C. Outputs
D. All of the above
E. None of the above

Correct Answer: D

Section:
Explanation:
The public Terraform Module Registry automatically exposes all the information about published modules, including required input variables, optional input variables and default values, and outputs. This helps users to understand how to use and configure the modules.

QUESTION 42
You must use different Terraform commands depending on the cloud provider you use.

A. True
B. False

Correct Answer: B
Section:
Explanation:
You do not need to use different Terraform commands depending on the cloud provider you use. Terraform commands are consistent across different providers, as they operate on the Terraform configuration files and state files, not on the provider APIs directly.

QUESTION 43
As a member of an operations team that uses infrastructure as code (lac) practices, you are tasked with making a change to an infrastructure stack running in a public cloud. Which pattern would follow laC best practices for making a change?

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

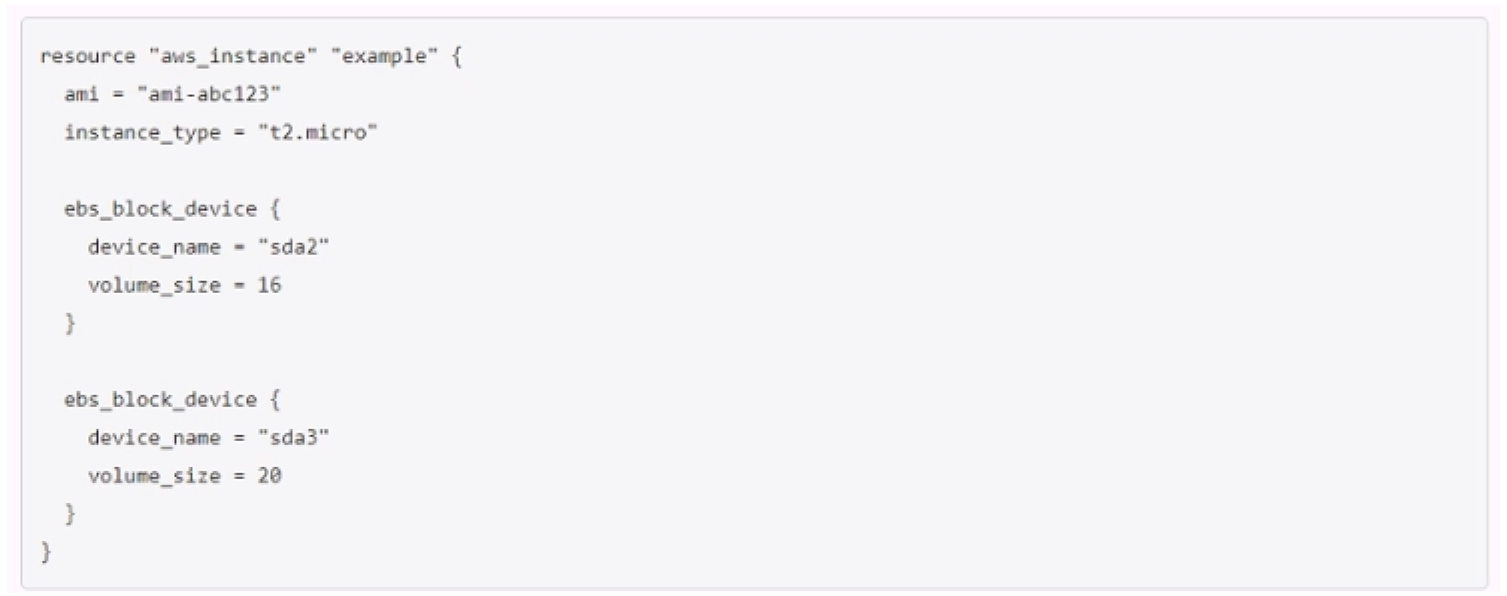
Correct Answer: E
Section:
Explanation:
You do not need to use different Terraform commands depending on the cloud provider you use. Terraform commands are consistent across different providers, as they operate on the Terraform configuration files and state files, not on the provider APIs directly.

QUESTION 44
You ate making changes to existing Terraform code to add some new infrastructure. When is the best time to run terraform validate?

A. After you run terraform apply so you can validate your infrastructure
B. Before you run terraform apply so you can validate your provider credentials
C. Before you run terraform plan so you can validate your code syntax
D. After you run terraform plan so you can validate that your state file is consistent with your infrastructure

Correct Answer: C
Section:
Explanation:
This is the best time to runterraform validate, as it will check your code for syntax errors, typos, and missing arguments before you attempt to create a plan. The other options are either incorrect or unnecessary.

QUESTION 45
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[sda2,sda3).volume_id
B. aws_Instance.example.ebs_block_device.[*].volume_id
C. aws_Instance.example.ebs_block_device.volume_ids
D. aws_instance.example-ebs_block_device.*.volume_id

Correct Answer: D
Section:
Explanation:
This is the correct way to reference the volume IDs associated with the ebs_block_device blocks in this configuration, using the splat expression syntax. The other options are either invalid or incomplete.

QUESTION 46
Which command should you run to check if all code in a Terraform configuration that references multiple modules is properly formatted without making changes?

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

Correct Answer: C
Section:
Explanation:
This command will check if all code in a Terraform configuration that references multiple modules is properly formatted without making changes, and will return a non-zero exit code if any files need formatting. The other commands will either make changes, list the files that need formatting, or not check the modules.

QUESTION 47
What kind of configuration block will create an infrastructure object with settings specified within the block?

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

Correct Answer: D
Section:
Explanation:
This is the kind of configuration block that will create an infrastructure object with settings specified within the block. The other options are not used for creating infrastructure objects, but for configuring providers, accessing state data, or querying data sources.

QUESTION 48
What is the Terraform style convention for indenting a nesting level compared to the one above it?

A. With a tab
B. With two spaces
C. With four spaces
D. With three spaces

Correct Answer: B
Section:
Explanation:
This is the Terraform style convention for indenting a nesting level compared to the one above it. The other options are not consistent with the Terraform style guide.

QUESTION 49
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 account. Execute terraform destroy and select the newly-created VM.
B. Delete the Terraform state file and execute terraform apply.
C. The Terraform state file only contains the one new VM. Execute terraform destroy.
D. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.

Correct Answer: C
Section:
Explanation:
This is the best way to delete the newly-created VM with Terraform, as it will only affect the resource that was created by your configuration and state file. The other options are either incorrect or inefficient.

QUESTION 50
When do changes invoked by terraform apply take effect?

A. After Terraform has updated the state file
B. Once the resource provider has fulfilled the request
C. Immediately
D. None of the above are correct

Correct Answer: B
Section:
Explanation:
Changes invoked byterraform applytake effect once the resource provider has fulfilled the request, not after Terraform has updated the state file or immediately. The state file is only a reflection of the real resources, not a source of truth.

QUESTION 51

What is the workflow for deploying new infrastructure with Terraform?

A. Write Terraform configuration, run terraform init to initialize the working directory or workspace, and run terraform apply
B. Write Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure
C. Write Terraform configuration, run terraform apply to create infrastructure, use terraform validate to confirm Terraform deployed resources correctly
D. Write Terraform configuration, run terraform plan to initialize the working directory or workspace, and terraform apply to create the infrastructure

Correct Answer: A
Section:
Explanation:
This is the workflow for deploying new infrastructure with Terraform, as it will create a plan and apply it to the target environment. The other options are either incorrect or incomplete.

QUESTION 52
Select the command that doesn't cause Terraform to refresh its state.

A. Terraform destroy
B. Terraform apply
C. Terraform plan
D. Terraform state list

Correct Answer: D
Section:
Explanation:
This is the command that does not cause Terraform to refresh its state, as it only lists the resources that are currently managed by Terraform in the state file. The other commands will refresh the state file before performing their operations, unless you use the-refresh=falseflag.

QUESTION 53
Which command lets you experiment with terraform expressions?

A. Terraform console
B. Terraform validate
C. Terraform env
D. Terraform test

Correct Answer: A
Section:
Explanation:
This is the command that lets you experiment with Terraform expressions, by providing an interactive console that allows you to evaluate expressions and see their results. You can use this command to test your expressions before using them in your configuration files.

QUESTION 54
If a DevOps team adopts AWS CloudFormation as their standardized method for provisioning public cloud resoruces, 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-native services
B. The organization decides to expand into Azure wishes to deploy new infrastructure
C. The team is asked to build a reusable code based that can deploy resources into any AWS region
D. The DevOps team is tasked with automating a manual, web console-based provisioning.

Correct Answer: B
Section:
Explanation:
This is the scenario that poses a challenge for this team, if they adopt AWS CloudFormation as their standardized method for provisioning public cloud resources, as CloudFormation only supports AWS services and resources, and cannot be used to provision infrastructure on other cloud platforms such as Azure.

QUESTION 55
When using a remote backend or terraform Cloud integration, where does Terraform save resource sate?

A. In an environment variable
B. On the disk
C. In the remote backend or Terraform Cloud
D. In memory

Correct Answer: C
Section:
Explanation:
This is where Terraform saves resource state when using a remote backend or Terraform Cloud integration, as it allows you to store and manage your state file in a remote location, such as a cloud storage service or Terraform Cloud's servers. This enables collaboration, security, and scalability for your Terraform infrastructure.

QUESTION 56
You add a new resource to an existing Terraform configuration, but do not update the version constraint in the configuration. The existing and new resources use the same provider. The working contains a .terraform.lock, hc1 file.
How will Terraform choose which version of the provider to use?

A. Terraform will use the version recorded in your lock file
B. Terraform will use the latest version of the provider for the new resource and the version recorded in the lock file to manage existing resources
C. Terraform will check your state file to determine the provider version to use
D. Terraform will use the latest version of the provider available at the time you provision your new resource

Correct Answer: A
Section:
Explanation:
This is how Terraform chooses which version of the provider to use, when you add a new resource to an existing Terraform configuration, but do not update the version constraint in the configuration. The lock file records the exact version of each provider that was installed in your working directory, and ensures that Terraform will always use the same provider versions until you runterraform init -upgradeto update them.

QUESTION 57
What is a key benefit of the Terraform state file?

A. A state file can schedule recurring infrastructure tasks
B. A state file is a source of truth for resources provisioned with Terraform
C. A state file is a source of truth for resources provisioned with a public cloud console
D. A state file is the desired state expressed by the Terraform code files

Correct Answer: B
Section:
Explanation:
This is a key benefit of the Terraform state file, as it stores and tracks the metadata and attributes of the resources that are managed by Terraform, and allows Terraform to compare the current state with the desired state

expressed by your configuration files.

QUESTION 58
You have to initialize a Terraform backend before it can be configured.

A. True
B. False

Correct Answer: B
Section:
Explanation:
You can configure a backend in your Terraform code before initializing it. Initializing a backend will store the state file remotely and enable features like locking and workspaces.Reference= [Terraform Backends]

QUESTION 59
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 state rm command to remove the VM from state file
B. Use the terraform taint command targeting the VMs then run terraform plan and terraform apply
C. Use the terraform apply command targeting the VM resources only
D. Delete the Terraform VM resources from your Terraform code then run terraform plan and terraform apply

Correct Answer: B
Section:
Explanation:
The terraform taint command marks a resource as tainted, which means it will be destroyed and recreated on the next apply. This way, you can replace the VM instance without affecting the database or other resources.Reference= [Terraform Taint]

QUESTION 60
How would you output returned values from a child module in the Terraform CLI output?

A. Declare the output in the root configuration
B. Declare the output in the child module
C. Declare the output in both the root and child module
D. None of the above

Correct Answer: C
Section:
Explanation:
To output returned values from a child module in the Terraform CLI output, you need to declare the output in both the child module and the root module. The child module output will return the value to the root module, and the root module output will display the value in the CLI.Reference= [Terraform Outputs]

QUESTION 61
If a module declares a variable with a default, that variable must also be defined within the module.

A. True
B. False

Correct Answer: B

Section:
Explanation:
A module can declare a variable with a default value without requiring the caller to define it. This allows the module to provide a sensible default behavior that can be customized by the caller if needed.Reference= [Module Variables]

QUESTION 62
A terraform apply can not __________ infrastructure.

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

Correct Answer: D
Section:
Explanation:
The terraform import command is used to import existing infrastructure into Terraform's state. This allows Terraform to manage and destroy the imported infrastructure as part of the configuration. The terraform import command does not modify the configuration, so the imported resources must be manually added to the configuration after the import.Reference= [Importing Infrastructure]

QUESTION 63
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

Correct Answer: D
Section:
Explanation:
The terraform import command is not part of any other Terraform workflow. It must be explicitly invoked by the user with the appropriate arguments, such as the resource address and the ID of the existing infrastructure to import.Reference= [Importing Infrastructure]

QUESTION 64
A provider configuration block is required in every Terraform configuration.
Example:

```
provider "provider_name" {
  ...
}
```

A. True
B. False

Correct Answer: B
Section:
Explanation:

A provider configuration block isnotrequired in every Terraform configuration. A provider configuration block can be omitted if its contents would otherwise be empty. Terraform assumes an empty default configuration for any provider that is not explicitly configured. However, some providers may require some configuration arguments (such as endpoint URLs or cloud regions) before they can be used. A provider's documentation should list which configuration arguments it expects.For providers distributed on the Terraform Registry, versioned documentation is available on each provider's page, via the ''Documentation'' link in the provider's header1.Reference= [Provider Configuration]1

QUESTION 65
You have multiple team members collaborating on infrastructure as code (IaC) using Terraform, and want to apply formatting standards for readability.
How can you format Terraform HCL (HashiCorp Configuration Language) code according to standard Terraform style convention?

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

Correct Answer: A
Section:
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. Running this command on your configuration files before committing them to source control can help ensure consistency of style between different Terraform codebases, and can also make diffs easier to read.You can also use the -check and -diff options to check if the files are formatted and display the formatting changes respectively2.Running the terraform fmt command during the code linting phase of your CI/CD process can help automate this process and enforce the formatting standards for your team.Reference= [Command: fmt]2

QUESTION 66
One remote backend configuration always maps to a single remote workspace.

A. True
B. False

Correct Answer: A
Section:
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 workspaces3. However, one remote backend configuration always maps to a single remote workspace, either by name or by prefix. You cannot use both name and prefix in the same backend configuration, or omit both.Doing so will result in a configuration error3.Reference= [Backend Type: remote]3

QUESTION 67
Which of the following does terraform apply change after you approve the execution plan? (Choose two.)

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

Correct Answer: A, D
Section:

Explanation:
Theterraform applycommand changes both the cloud infrastructure and the state file after you approve the execution plan. The command creates, updates, or destroys the infrastructure resources to match the configuration. It also updates the state file to reflect the new state of the infrastructure. The.terraformdirectory, the execution plan, and the Terraform code are not changed by theterraform applycommand.Reference=Command: applyandPurpose of Terraform State

QUESTION 68
You are working on some new application features and you want to spin up a copy of your production deployment to perform some quick tests. In order to avoid having to configure a new state backend, what open source Terraform feature would allow you create multiple states but still be associated with your current code?

A. Terraform data sources
B. Terraform local values
C. Terraform modules
D. Terraform workspaces
E. None of the above

Correct Answer: D
Section:
Explanation:
Terraform workspaces allow you to create multiple states but still be associated with your current code. Workspaces are like ''environments'' (e.g. staging, production) for the same configuration. You can use workspaces to spin up a copy of your production deployment for testing purposes without having to configure a new state backend. Terraform data sources, local values, and modules are not features that allow you to create multiple states.Reference=WorkspacesandHow to Use Terraform Workspaces

QUESTION 69
Which of the following commands would you use to access all of the attributes and details of a resource managed by Terraform?

A. terraform state list 'provider_type.name'
B. terraform state show 'provider_type.name'
C. terraform get 'provider_type.name'
D. terraform state list

Correct Answer: B
Section:
Explanation:
Theterraform state showcommand allows you to access all of the attributes and details of a resource managed by Terraform. You can use the resource address (e.g.provider_type.name) as an argument to show the information about a specific resource. Theterraform state listcommand only shows the list of resources in the state, not their attributes. Theterraform getcommand downloads and installs modules needed for the configuration. It does not show any information about resources.Reference= [Command: state show] and [Command: state list]

QUESTION 70
If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?

A. Run terraform refresh
B. It will happen automatically
C. Manually update the state fire
D. Run terraform import

Correct Answer: B
Section:
Explanation:

If you manually destroy infrastructure, Terraform will automatically detect the change and update the state file during the next plan or apply. Terraform compares the current state of the infrastructure with the desired state in the configuration and generates a plan to reconcile the differences. If a resource is missing from the infrastructure but still exists in the state file, Terraform will attempt to recreate it. If a resource is present in the infrastructure but not in the state file, Terraform will ignore it unless you use the terraform import command to bring it under Terraform's management.Reference= [Terraform State]

QUESTION 71
Which option cannot be used to keep secrets out of Terraform configuration files?

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

Correct Answer: D
Section:
Explanation:
A secure string is not a valid option to keep secrets out of Terraform configuration files. A secure string is a feature of AWS Systems Manager Parameter Store that allows you to store sensitive data encrypted with a KMS key. However, Terraform does not support secure strings natively and requires a custom data source to retrieve them. The other options are valid ways to keep secrets out of Terraform configuration files. A Terraform provider can expose secrets as data sources that can be referenced in the configuration. Environment variables can be used to set values for input variables that contain secrets. A -var flag can be used to pass values for input variables that contain secrets from the command line or a file.Reference= [AWS Systems Manager Parameter Store], [Terraform AWS Provider Issue #55], [Terraform Providers], [Terraform Input Variables]

QUESTION 72
When using Terraform to deploy resources into Azure, which scenarios are true regarding state files? (Choose two.)

A. When you change a Terraform-managed resource via the Azure Cloud Console, Terraform updates the state file to reflect the change during the next plan or apply
B. Changing resources via the Azure Cloud Console records the change in the current state file
C. When you change a resource via the Azure Cloud Console, Terraform records the changes in a new state file
D. Changing resources via the Azure Cloud Console does not update current state file

Correct Answer: A, D
Section:
Explanation:
Terraform state is a representation of the infrastructure that Terraform manages. Terraform uses state to track the current status of the resources it creates and to plan future changes. However, Terraform state is not aware of any changes made to the resources outside of Terraform, such as through the Azure Cloud Console, the Azure CLI, or the Azure API. Therefore, changing resources via the Azure Cloud Console does not update the current state file, and it may cause inconsistencies or conflicts with Terraform's desired configuration. To avoid this, it is recommended to manage resources exclusively through Terraform or to use theterraform importcommand to bring existing resources under Terraform's control.
When you change a Terraform-managed resource via the Azure Cloud Console, Terraform does not immediately update the state file to reflect the change. However, the next time you runterraform planorterraform apply, Terraform will compare the state file with the actual state of the resources in Azure and detect any drifts or differences. Terraform will then update the state file to match the current state of the resources and show you the proposed changes in the execution plan. Depending on the configuration and the change, Terraform may try to undo the change, modify the resource further, or recreate the resource entirely. To avoid unexpected or destructive changes, it is recommended to review the execution plan carefully before applying it or to use theterraform refreshcommand to update the state file without applying any changes.
Reference=Purpose of Terraform State,Terraform State,Managing State,Importing Infrastructure, [Command: plan], [Command: apply], [Command: refresh]

QUESTION 73
Which of the following should you put into the required_providers block?

A. version >= 3.1
B. version = ">= 3.1"
C. version ~> 3.1

Correct Answer: B
Section:
Explanation:
Therequired_providersblock is used to specify the provider versions that the configuration can work with. Theversionargument accepts a version constraint string, which must be enclosed in double quotes. The version constraint string can use operators such as>=,~>,=, etc. to specify the minimum, maximum, or exact version of the provider. For example,version = '>= 3.1'means that the configuration can work with any provider version that is 3.1 or higher.Reference= [Provider Requirements] and [Version Constraints]