HashiCorp.Terraform-Associate-003.v2026-04-24.q106

Exam Code:	Terraform-Associate-003
Exam Name:	HashiCorp Certified: Terraform Associate (003) (HCTA0-003)
Certification Provider:	HashiCorp
Free Question Number:	106
Version:	v2026-04-24
# of views:	111
# of Questions views:	1060
https://www.freepdfdumps.com/HashiCorp.Terraform-Associate-003.v2026-04-24.q106.html	

NEW QUESTION: 1
What type of block is used to construct a collection of nested configuration blocks?
A. Dynamic
B. For_each
C. Nesting
D. repeated.
Answer: A (LEAVE A REPLY)
This is the type of block that is used to construct a collection of nested configuration blocks, by using a for_each argument 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.

NEW QUESTION: 2
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
Answer: D (LEAVE A REPLY)
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=false flag.

NEW QUESTION: 3
You have deployed a new webapp with a public IP address on a cloud provider. However, you did not create any outputs for your code. What is the best method to quickly find the IP address of the resource you deployed?

A. In a new folder, use the terraform_remote_state data source to load in the state file, then write an output for each resource that you find the state file
B. Run terraform state list to find the name of the resource, then terraform state show to find the attributes including public IP address
C. Run terraform output ip_address to view the result
D. Run terraform destroy then terraform apply and look for the IP address in stdout
Answer: B (LEAVE A REPLY)
This is a quick way to inspect the state file and find the information you need without modifying anything5.
The other options are either incorrect or inefficient.

NEW QUESTION: 4
Which type of block fetches or computes information for use elsewhere in a Terraform configuration?
A. data
B. local
C. resource
D. provider
Answer: (SHOW ANSWER)
In Terraform, adatablock is used to fetch or compute information from external sources for use elsewhere in the Terraform configuration. Unlikeresourceblocks that manage infrastructure,datablocks gather information without directly managing any resources. This can include querying for data from cloud providers, external APIs, or other Terraform states.References= This definition and usage ofdatablocks are covered in Terraform's official documentation, highlighting their role in fetching external information to inform Terraform configurations.

NEW QUESTION: 5
Which command must you first run before performing further Terraform operations in a working directory?
A. terraform import
B. terraform workspace
C. terraform plan
D. terraform init
Answer: (SHOW ANSWER)
terraform init is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It initializes a working directory containing Terraform configuration files and downloads any required providers and modules. The other commands are used for different purposes, such as importing existing resources, switching between workspaces, generating execution plans, etc.

NEW QUESTION: 6
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
Answer: (SHOW ANSWER)
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.
References = This information can be verified from the official Terraform documentation on the terraform init command provided by HashiCorp Developer1.

NEW QUESTION: 7
Any user can publish modules to the public Terraform Module Registry.
A. True
B. False
Answer: A (LEAVE A REPLY)
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.
References = The information can be verified from the Terraform Registry documentation on Publishing Modules provided by HashiCorp Developer1.

NEW QUESTION: 8
A Terraform output that sets the "sensitive" argument to true will not store that value in the state file.
A. True
B. False
Answer: (SHOW ANSWER)
From the Terraform Output Docs - Sensitive:
"Marking an output as sensitive prevents Terraform from showing its value in the CLI output,but it will still bestored in the state file." Thus, it protects display, not storage.

NEW QUESTION: 9

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 afterward 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 (LEAVE A REPLY)

Terraform follows adeclarativeapproach, meaning it ensures the infrastructure matches the configuration.

If you run terraform apply againwithout any changes, Terraform willdetect that the infrastructure is already up to dateand willdo nothing.

The command will complete successfully with the message"No changes. Your infrastructure matches the configuration." Official Terraform Documentation Reference:

Terraform Apply - HashiCorp Documentation

NEW QUESTION: 10

You just upgraded the version of a provider in an existing Terraform project. What do you need to do to install the new provider?

A. Run terraform refresh.

B. Run terraform init -upgrade.

C. Run terraform apply -upgrade.

D. Upgrade your version of Terraform.

Answer: B (LEAVE A REPLY)

When upgrading a provider,you must run terraform init -upgrade to install the new version.

* Thisdownloads the latest provider versionand updates the local dependency cache.
* terraform refresh(A)only updates the state file but doesnotupgrade providers.
* terraform apply -upgrade(C)isnot a valid command.
* terraform version can be upgraded separately, but(D)does not install a new provider version.

Official Terraform Documentation Reference:

Upgrading Providers in Terraform

NEW QUESTION: 11

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

A. version >= 3.1

B. version = ">= 3.1"

C. version ~> 3.1

Answer: B (LEAVE A REPLY)

The required_providers block is used to specify the provider versions that the configuration can work with.

The version argument 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. References = [Provider Requirements] and [Version Constraints]

NEW QUESTION: 12
Only the user that generated a plan may apply it.
A. True
B. False
Answer: B (LEAVE A REPLY)
Any user with permission to apply a plan can apply it, not only the user that generated it. This allows for collaboration and delegation of tasks among team members.

NEW QUESTION: 13
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
Answer: B (LEAVE A REPLY)
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. References = [Terraform Taint]

NEW QUESTION: 14
If one of your modules uses a local value, you can expose that value to callers of the module by defining a Terraform output in the module's configuration.
A. True
B. False
Answer: A (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (True):Local values (locals {}) are scoped to a module and are not directly visible outside. To make them available to callers, you must define outputs in the module. Outputs act as the interface for exposing values from a child module to the root module.
* Analysis of Incorrect Option:
* False: Incorrect, because locals cannot be exposed directly; only via outputs.

* Key Concept:Outputs are the way to expose information outside of a module.
Reference:Terraform Exam Objective - Interact with Terraform Modules.

NEW QUESTION: 15
What is the purpose of the terraform.lock.hcl file in Terraform?
A. There is no such file.
B. Storing references to workspaces, which are locked.
C. Preventing Terraform runs from occurring.
D. Tracking specific provider dependencies.
Answer: D (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (D):The terraform.lock.hcl file is automatically created and maintained by Terraform. It records the specific versions and checksums of providers used in a configuration, ensuring consistent runs across environments and machines.
* Analysis of Incorrect Options:
* A. There is no such file: Incorrect - the lock file exists and is crucial for dependency management.
* B. Storing references to workspaces: Incorrect - workspaces are tracked in the state file, not the lock file.
* C. Preventing Terraform runs: Incorrect - the lock file does not block runs; it enforces consistent provider versions.
* Key Concept:The terraform.lock.hcl file enforces provider version consistency, which is critical for reproducible and reliable Terraform deployments.
Reference:Terraform Exam Objective - Read, Generate, and Modify Configurations.

NEW QUESTION: 16
Running terraform fmt without any flags in a directory with Terraform configuration files check the formatting of those files without changing their contents.
A. True
B. False
Answer: B (LEAVE A REPLY)
Running terraform fmt without any flags in a directory with Terraform configuration files will not check the formatting of those files without changing their contents, but will actually rewrite them to a canonical format and style. If you want to check the formatting without making changes, you need to use the -check flag.

NEW QUESTION: 17
Which parameters does terraform import require? Choose two correct answers.
A. Provider
B. Resource ID
C. Resource address
D. Path
Answer: (SHOW ANSWER)
These are the parameters that terraform import requires, as they allow Terraform to identify the existing resource that you want to import into your state file, and match it with the corresponding configuration block in your files.

NEW QUESTION: 18
What is the provider for the resource shown in the Exhibit?
resource "aws_vpc" "main" {
name = "test"
}
A. VPC
B. test
C. main
D. aws
Answer: D (LEAVE A REPLY)
The provider for the aws_vpc resource is aws, as the resource type begins with aws_, which denotes that it is managed by the AWS provider.
References:
Terraform Providers

NEW QUESTION: 19
Terraform configuration (including any module references) can contain only one Terraform provider type.
A. True
B. False
Answer: B (LEAVE A REPLY)
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. References = [Providers], [Provider Requirements], [Provider Configuration]

NEW QUESTION: 20
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"
Answer: A (LEAVE A REPLY)
The module source path that does not specify a remote module is source = "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. References = [Module Sources], [Local Paths], [Terraform Registry], [Generic Git Repository],
[GitHub]

NEW QUESTION: 21
When you use a backend that requires authentication, it is best practice to:
A. Run all Terraform commands on a shared server or container.
B. Configure the authentication credentials in your Terraform configuration files, and store them in version control.
C. Use environment variables to configure authentication credentials outside of your Terraform configuration.
D. None of the above.
Answer: (SHOW ANSWER)
Detailed Explanation:
* Rationale for Correct Answer (C):Credentials should not be hardcoded in Terraform files. Best practice is to configure them via environment variables or secret managers.
* Analysis of Incorrect Options:
* A: Shared servers introduce risk and drift.
* B: Storing secrets in config/version control is insecure.
* D: Incorrect since option C is valid.
* Key Concept:Separation of credentials from code is a Terraform security best practice.
Reference:Terraform Exam Objective - Navigate Terraform State and Backends.

NEW QUESTION: 22

You ate creating a Terraform configuration which needs to make use of multiple providers, one for AWS and one for Datadog. Which of the following provider blocks would allow you to do this?

A)

B)

C)

D)

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C (LEAVE A REPLY)

Option C is the correct way to configure multiple providers in a Terraform configuration. Each provider block must have a name attribute that specifies which provider it configures2. The other options are either missing the name attribute or using an invalid syntax.

NEW QUESTION: 23

Which of the following is not a valid string function in Terraform?

A. chomp

B. join

C. slice

D. split

Answer: C (LEAVE A REPLY)

Referencing Terraform Built-in Functions:

* #chomp: removes trailing newlines
* #join: combines list into string
* #split: splits string into list
* #slice:is nota valid Terraform string function (it's used in other languages like Python)

NEW QUESTION: 24
In a HCP Terraform/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: A (LEAVE A REPLY)
Speculative Plans: Terraform Cloud's speculative plan feature runs automatically when changes are detected in a linked VCS repository, enabling users to review potential infrastructure changes without committing them.
Automatic Integration: This feature automates the planning process by triggering when changes are committed, aiding teams in previewing infrastructure changes seamlessly.
For further understanding, see the Terraform Cloud VCS Integration documentation.

NEW QUESTION: 25
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
Answer: C (LEAVE A REPLY)
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.

NEW QUESTION: 26
Which Terraform collection type should you use to store key/value pairs?
A. Set
B. Map
C. Tuple
D. list
Answer: B (LEAVE A REPLY)
The Terraform collection type that should be used to store key/value pairs is map. A map is a collection of values that are accessed by arbitrary labels, called keys. The keys and values can be of any type, butthe keys must be unique within a map. For example, var = { key1 = "value1", key2 = "value2" } is a map with two key
/value pairs. Maps are useful for grouping related values together, such as configuration options or metadata. References = [Collection Types], [Map Type Constraints]

NEW QUESTION: 27
Which two steps are required to provision new infrastructure in the Terraform workflow?
(Pick the 2 correct responses below)

A. import
B. plan
C. validate
D. init
E. apply
Answer: (SHOW ANSWER)
Detailed Explanation:
* Rationale for Correct answer:
* init: Initializes the working directory, downloads providers and modules.
* apply: Executes the planned changes and provisions infrastructure.
Both steps are mandatory for provisioning new infrastructure.
* Analysis of Incorrect Options (Distractors):
* A. import: Used for existing resources, not provisioning new ones.
* B. plan: Optional but recommended; not strictly required.
* C. validate: Checks syntax only; does not provision resources.
* Key Concept:The Terraform workflow requires initialization and application to create infrastructure.
Reference:Terraform Exam Objective - Understand Terraform Basics and CLI

NEW QUESTION: 28
Terraform installs its providers during which phase?
A. Plan
B. Init
C. Refresh
D. All of the above
Answer: B (LEAVE A REPLY)
Terraforminstalls providers during the terraform init phase.
Providers aredownloaded and installedwhen running terraform init.
A (terraform plan)is incorrect because terraform plan only calculates changes but does not install providers.
C (terraform refresh)is incorrect because terraform refresh only updates the state but does not install providers.
Official Terraform Documentation Reference:
Provider Installation - HashiCorp Documentation

NEW QUESTION: 29
A resource block is shown in the Exhibit space of this page. What is the Terraform resource name of the resource block?
A. test
B. google
C. compute_instance

D. main

Answer: D (LEAVE A REPLY)

In Terraform, theresource nameis the second argument in the resource block.

The structure of a resource block is:

resource "provider_resource_type" "resource_name" {

Configuration settings

}

Here, the provider type isgoogle_compute_instance, and theresource nameis "main".

The name "test" is simply the value assigned to the name attribute, which is unrelated to the Terraform resource name.

"google" and "compute_instance" are part of the provider and resource type, not the resource name.

Official Terraform Documentation Reference:

Terraform Resource Documentation

NEW QUESTION: 30

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

Answer: E (LEAVE A REPLY)

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 named terraform.tfstate, which can be used to track and manage the state of your infrastructure.

NEW QUESTION: 31

You want to define multiple data disks as nested blocks inside the resource block for a virtual machine. What Terraform feature would help you define the blocks using the values in a variable?

A. Local values

B. Count arguments

C. Collection functions

D. Dynamic blocks

Answer: D (LEAVE A REPLY)

Dynamic blocks in Terraform allow you to define multiple nested blocks within a resource based on the values of a variable. This feature is particularly useful for scenarios where the number of nested blocks is not fixed and can change based on variable input.

NEW QUESTION: 32
What are some benefits of using Sentinel with Terraform Cloud/Terraform Cloud? Choose three correct answers.
A. You can restrict specific resource configurations, such as disallowing the use of CIDR=0.0.0.0/0.
B. You can check out and check in cloud access keys
C. Sentinel Policies can be written in HashiCorp Configuration Language (HCL)
D. Policy-as-code can enforce security best practices
E. You can enforce a list of approved AWS AMIs
Answer: A,D,E (LEAVE A REPLY)
Sentinel is a policy-as-code framework that allows you to define and enforce rules on your Terraform configurations, states, and plans1. Some of the benefits of using Sentinel with Terraform Cloud/Terraform Enterprise are:
* You can restrict specific resource configurations, such as disallowing the use of CIDR=0.0.0.0/0, which would open up your network to the entire internet. This can help you prevent misconfigurations or security vulnerabilities in your infrastructure2.
* Policy-as-code can enforce security best practices, such as requiring encryption, authentication, or compliance standards. This can help you protect your data and meet regulatory requirements3.
* You can enforce a list of approved AWS AMIs, which are pre-configured images that contain the operating system and software you need to run your applications. This can help you ensure consistency, reliability, and performance across your infrastructure4.
References =
* 1: Terraform and Sentinel | Sentinel | HashiCorp Developer
* 2: Terraform Learning Resources: Getting Started with Sentinel in Terraform Cloud
* 3: Exploring the Power of HashiCorp Terraform, Sentinel, Terraform Cloud ...
* 4: Using New Sentinel Features in Terraform Cloud - Medium

NEW QUESTION: 33
Which Terraform command checks that your configuration syntax is correct?
A. terraform validate
B. terraform init
C. terraform show

D. terraform fmt

Answer: A (LEAVE A REPLY)

The terraform validate command is used to check that your Terraform configuration files are syntactically valid and internally consistent. It is a useful command for ensuring your Terraform code is error-free before applying any changes to your infrastructure.

NEW QUESTION: 34

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

A. True

B. False

Answer: B (LEAVE A REPLY)

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. References = [Terraform Backends]

NEW QUESTION: 35

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

Answer: C (LEAVE A REPLY)

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. References = [Terraform Outputs]

NEW QUESTION: 36

Infrastructure as Code (IaC) can be stored in a version control system along with application code.

A. True

B. False

Answer: A (LEAVE A REPLY)

Infrastructure as Code (IaC) can indeed be stored in a version control system along with application code.

This practice is a fundamental principle of modern infrastructure management, allowing teams to apply software development practices like versioning, peer review, and CI/CD to infrastructure management.

Storing IaC configurations in version control facilitates collaboration, history tracking, and change management.References = While this concept is a foundational aspect of IaC and is widely accepted in the industry, direct references from the HashiCorp Terraform Associate (003) study

materials were not found in the provided files. However, this practice is encouraged in Terraform's best practices and various HashiCorp learning resources.

NEW QUESTION: 37
Which argument can you use toprevent unexpected updatesto a module's configuration when calling Terraform Registry modules?
A. source
B. count
C. version
D. lifecycle
Answer: (SHOW ANSWER)
Theversion argumentin a module ensures Terraform uses aspecific versionof a module, preventing unintended updates.
A (source)- Specifies the module source butdoes not control versioning.
B (count)- Controls how many instances of a resource/module exist,not updates.
D (lifecycle)- Controls how resources behavebut does not control module versioning.
Official Terraform Documentation Reference:
Module Version Constraints

NEW QUESTION: 38
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
Answer: A (LEAVE A REPLY)
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.

NEW QUESTION: 39
Before you can use a remote backend, you must first execute terra-form init.
A. True
B. False
Answer: A (LEAVE A REPLY)
Before using a remote backend in Terraform, it is mandatory to runterraform init. This command initializes a Terraform working directory, which includes configuring the backend. If aremote backend is specified, terraform initwill set up the working directory to use it, including copying any existing state to the remote backend if necessary.References= This principle is a fundamental part of working with Terraform and its backends, as outlined in general Terraform documentation

and best practices. The specific HashiCorp Terraform Associate (003) study materials in the provided files did not include direct references to this information.

NEW QUESTION: 40
Which type of block fetches or computes information for use elsewhere in a Terraform configuration?
A. data
B. local
C. resource
D. provider
Answer: A (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (A):data blocks fetch information from providers (like AMI IDs, network info) that can be used in other resources.
* Analysis of Incorrect Options:
* B. local: Stores values, doesn't fetch external data.
* C. resource: Defines managed infrastructure, not read-only data.
* D. provider: Configures connection to APIs, not data fetching.
* Key Concept:Data sources are read-only queries into existing infrastructure.
Reference:Terraform Exam Objective - Read, Generate, and Modify Configurations.

NEW QUESTION: 41
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 runfirst?
A. terraform apply
B. terraform init
C. terraform plan
D. terraform show
Answer: B (LEAVE A REPLY)
B (terraform init)-Must be run firstto initialize the Terraform working directory, download providers, and configure the backend.
A (terraform apply)- Requires initialization first, so itcannotbe run before terraform init.
C (terraform plan)- Also requires terraform init first to generate a plan.
D (terraform show)- Displays the state,not relevantfor first-time deployment.
Official Terraform Documentation Reference:
terraform init - HashiCorp Documentation

NEW QUESTION: 42
What does the default "local" Terraform backend store?
A. tfplan files

B. State file
C. Provider plugins
D. Terraform binary
Answer: (SHOW ANSWER)
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.

NEW QUESTION: 43
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
Answer: D (LEAVE A REPLY)
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.

NEW QUESTION: 44
What does state looking accomplish?
A. Prevent accidental Prevent accident deletion of the state file
B. Blocks Terraform commands from modifying, the state file
C. Copies the state file from memory to disk
D. Encrypts any credentials stored within the state file
Answer: (SHOW ANSWER)
This is what state locking accomplishes, by preventing other users from modifying the state file while a Terraform operation is in progress. This prevents conflicts and data loss.

NEW QUESTION: 45
Which command doesnotcause Terraform to refresh its state?
A. terraform state list
B. terraform plan
C. terraform apply
D. terraform destroy
Answer: A (LEAVE A REPLY)
terraform state listonly displays resources stored in the state filebut doesnot interact with the cloud provideror refresh the state.
terraform plan, terraform apply, and terraform destroycompare or modify the infrastructure, so theyrefresh the stateto ensure accuracy.

Official Terraform Documentation Reference:
terraform state list - HashiCorp Documentation

NEW QUESTION: 46
You modified your Terraform configuration to fix a typo in the resource ID by renaming it from photoes to photos. What configuration will you add to update the resource ID in state without destroying the existing resource?
Original configuration:
resource "aws_s3_bucket" "photoes" {
bucket_prefix = "images"
}
Updated configuration:
resource "aws_s3_bucket" "photos" {
bucket_prefix = "images"
}
A. moved {
from = aws_s3_bucket.photoes
to = aws_s3_bucket.photos
}
B. moved {
bucket.photoes = aws_s3_bucket.photos
}
C. moved {
aws_s3_bucket.photoes = aws_s3_bucket.photos
}
D. None. Terraform will automatically update the resource ID.
Answer: (SHOW ANSWER)
Detailed Explanation:
* Rationale for Correct Answer (A):Terraform does not automatically update state references when resource identifiers are renamed. Instead, you must use a moved block in your configuration to inform Terraform how to map the old resource to the new one. This prevents Terraform from destroying and recreating the resource.
* Analysis of Incorrect Options:
* B & C: Incorrect syntax - moved requires from and to.
* D: Incorrect, Terraform won't auto-detect renames and will plan to destroy and recreate the resource unless a moved block is provided.
* Key Concept:The moved block is essential for refactoring resource names without losing resources in state.
Reference:Terraform Exam Objective - Implement and Maintain State.

NEW QUESTION: 47
Which configuration consistency errors does terraform validate report?
A. Terraform module isn't the latest version
B. Differences between local and remote state
C. Declaring a resource identifier more than once
D. A mix of spaces and tabs in configuration files
Answer: C (LEAVE A REPLY)
Terraform validate reports configuration consistency errors, such as declaring a resource identifier more than once. This means that the same resource type and name combination is used for multiple resource blocks, which is not allowed in Terraform. For example, resource "aws_instance" "example" {...} cannot be used more than once in the same configuration. Terraform validate does not report errors related to module versions, state differences, or formatting issues, as these are not relevant for checking the configuration syntax and structure. References = [Validate Configuration], [Resource Syntax]

NEW QUESTION: 48
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 }
Answer: B,C (LEAVE A REPLY)
These are two ways to produce a list of the IDs from a list of objects that have an attribute id, using either a for expression or a splat expression syntax.

NEW QUESTION: 49
All modules published on the official Terraform Registry are maintained and tested by HashiCorp.
A. True
B. False
Answer: B (LEAVE A REPLY)
Detailed Explanation:

* Rationale for Correct answer:While HashiCorp publishes and maintains some official modules, many modules on the Terraform Registry are community-maintained. HashiCorp does not test or maintain all published modules. The Registry clearly distinguishes between official, partner, and community modules.
* Analysis of Incorrect Options (Distractors):
* A. True: Incorrect because not all Registry modules are maintained or tested by HashiCorp.
* Key Concept:Understanding the difference between official, partner, and community modules in the Terraform Registry.
Reference:Terraform Exam Objective - Interact with Terraform Modules

NEW QUESTION: 50
Variables declared within a module are accessible outside of the module.
A. True
B. False
Answer: B (LEAVE A REPLY)
Variables declared within a module are only accessible within that module, unless they are explicitly exposed as output values1.

NEW QUESTION: 51
How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? Choose two correct answers.
A. End-users have to request infrastructure changes
B. Ticket based systems generate a full audit trail of the request and fulfillment process
C. Users can access catalog of approved resources from drop down list in a request form
D. The more resources your organization needs, the more tickets your infrastructure team has to process
Answer: (SHOW ANSWER)
These are some of the ways that a ticket-based system can slow down infrastructure provisioning and limit the ability to scale, as they introduce delays, bottlenecks, and manual interventions in the process of creating and modifying infrastructure.

NEW QUESTION: 52
Which of these ate features of Terraform Cloud? Choose two correct answers.
A. Automated infrastructure deployment visualization
B. Automatic backups
C. A web-based user interface (UI)
D. Remote state storage
Answer: C,D (LEAVE A REPLY)
These are features of Terraform Cloud, which is a hosted service that provides a web-based UI, remote state storage, remote operations, collaboration features, and more for managing your Terraform infrastructure.

NEW QUESTION: 53
What does this code do?
terraform { required_providers { aws = ">= 3.0" }}
A. Requires any version of the AWS provider > = 3.0 and <4.0
B. Requires any version of the AWS provider >= 3.0
C. Requires any version of the AWS provider > = 3.0 major release. like 4.1
D. Requires any version of the AWS provider > 3.0
Answer: A (LEAVE A REPLY)
From the Terraform Provider Requirements:
">= 3.0" means any versiongreater than or equal to 3.0- including 4.x and beyond.
A (wrong): Would need >= 3.0, < 4.0
C/D (wrong): > excludes 3.0 - not what's written.

NEW QUESTION: 54
You are tasked with making a change to an infrastructure stack running in a public cloud using HCP Terraform/Terraform Cloud. Which pattern follows IaC best practices?
A. Make the change via the public cloud API endpoint.
B. Submit a pull request and wait for an approved merge of the proposed changes.
C. Clone the repository containing your infrastructure code and then run the code.
D. Use the public cloud console to make the change after approval.
E. Make the change programmatically via the cloud CLI.
Answer: B (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (B):IaC best practice is to manage infrastructure through version-controlled code. Changes should be reviewed and approved (via PRs), ensuring collaboration, traceability, and automation.
* Analysis of Incorrect Options:
* A, D, E: Making direct/manual changes bypasses IaC practices and causes drift.
* C: Running code without PR review skips collaboration and approval.
* Key Concept:Infrastructure as Code emphasizes version control + peer review + automation.
Reference:Terraform Exam Objective - Understand Infrastructure as Code (IaC) Concepts.

NEW QUESTION: 55
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
Answer: A (LEAVE A REPLY)
You should run terraform fmt to rewrite all Terraform configurations within the current working directory to conform to Terraform-style conventions. This command applies a subsetof 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. References = : Command: fmt : Using Terraform fmt Command to Format Your Terraform Code

NEW QUESTION: 56
When does Sentinel enforce policy logic during a Terraform Cloud run?
A. Before the plan phase
B. During the plan phase
C. Before the apply phase
D. After the apply phase
Answer: C (LEAVE A REPLY)
Sentinel policies are checked after the plan stage of a Terraform run, but before it can be confirmed or the terraform apply is executed3. This allows you to enforce rules on your infrastructure before it is created or modified.

NEW QUESTION: 57
What is terraform plan -refresh-only intended to detect?
A. Terraform configuration code changes
B. Corrupt state files
C. State file drift
D. Empty state files
Answer: C (LEAVE A REPLY)
The terraform refresh-only command is intended to detect state file drift. This command synchronizes the state file with the actual infrastructure, updating the state to reflect any changes that have occurred outside of Terraform.
From terraform plan -refresh-only:
"Use this to detectdrift, i.e., changes made outside of Terraform."
It compares actual infrastructure to what's recorded in the state file.

NEW QUESTION: 58
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
Answer: A,D (LEAVE A REPLY)

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 the terraform import command 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 run terraform plan or terraform 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 the terraform refresh command to update the state file without applying any changes.
References = Purpose of Terraform State, Terraform State, Managing State, Importing Infrastructure,
[Command: plan], [Command: apply], [Command: refresh]

NEW QUESTION: 59
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
Answer: (SHOW ANSWER)
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. References =
[AWS Systems Manager Parameter Store], [Terraform AWS Provider Issue #55], [Terraform Providers],
[Terraform Input Variables]

NEW QUESTION: 60
When do you need to explicitly execute Terraform in refresh-only mode?
A. Before every terraform plan.
B. Before every terraform apply.
C. Before every terraform import.
D. None of the above.
Answer: C (LEAVE A REPLY)
Purpose of Refresh-Only Mode: Running Terraform in refresh-only mode updates Terraform's state file with the current state of resources in your infrastructure without making changes to the resources themselves.
Context of Terraform Import: When using terraform import, you're adding existing resources to the state file, and running Terraform in refresh-only mode before this operation can ensure that any initial configuration syncs precisely with the actual state.
For more on refresh-only mode in relation to terraform import, refer to Terraform's import documentation.

NEW QUESTION: 61
When using multiple configuration of the same Terraform provider, what meta-argument must you include in any non-default provider configurations?
A. Alias
B. Id
C. Depends_on
D. name
Answer: A (LEAVE A REPLY)
This is the meta-argument that you must include in any non-default provider configurations, as it allows you to give a friendly name to the configuration and reference it in other parts of your code. The other options are either invalid or irrelevant for this purpose.

NEW QUESTION: 62

When you include a module block in your configuration that references a module from the Terraform Registry, the "version" attribute is required.

A. True

B. False

Answer: B (LEAVE A REPLY)

The "version" attribute is optional when referencing a module from the Terraform Registry. If not specified, the latest version will be used, but it is often recommended to specify a version to ensure consistency across environments.

References:

* Terraform Module Versioning

NEW QUESTION: 63

Which of these statements about Terraform Cloud workspaces is false?

A. They have role-based access controls

B. You must use the CLI to switch between workspaces

C. Plans and applies can be triggered via version control system integrations

D. They can securely store cloud credentials

Answer: (SHOW ANSWER)

The statement that you must use the CLI to switch between workspaces is false. Terraform Cloud workspaces are different from Terraform CLI workspaces. Terraform Cloud workspaces are required and represent all of the collections of infrastructure in an organization. They are also a major component of role-based access in Terraform Cloud. You can grant individual users and user groupspermissions for one or more workspaces that dictate whether they can manage variables, perform runs, etc. You can create, view, and switch between Terraform Cloud workspaces using the Terraform Cloud UI, the Workspaces API, or the Terraform Enterprise Provider5. Terraform CLI workspaces are optional and allow you to create multiple distinct instances of a single configuration within one working directory. They are useful for creating disposable environments for testing or experimenting without affecting your main or production environment. You can create, view, and switch between Terraform CLI workspaces using the terraform workspace command6. The other statements about Terraform Cloud workspaces are true. They have role-based access controls that allow you to assign permissions to users and teams based on their roles and responsibilities. You can create and manage roles using the Teams API or the Terraform Enterprise Provider7. Plans and applies can be triggered via version control system integrations that allow you to link your Terraform Cloud workspaces to your VCS repositories. You can configure VCS settings, webhooks, and branch tracking to automate your Terraform Cloud workflow8. They can securely store cloud credentials as sensitive variables that are encrypted at rest and only decrypted when needed. You can manage variables using the Terraform Cloud UI, the Variables API, or the Terraform Enterprise Provider9. References = [Workspaces]5, [Terraform CLI Workspaces]6,
[Teams and Organizations]7, [VCS Integration]8, [Variables]9

NEW QUESTION: 64
All standard backend types support state locking, and remote operations like plan, apply, and destroy.
A. True
B. False
Answer: B (LEAVE A REPLY)
Not all standard backend types support state locking and remote operations like plan, apply, and destroy. For example, the local backend does not support remote operations and state locking. State locking is a feature that ensures that no two users can make changes to the state file at the same time, which is crucial for preventing race conditions. Remote operations allow running Terraform commands on a remote server, which is supported by some backends like remote or consul, but not all.
References:
Terraform documentation on backends: Terraform Backends
Detailed backend support: Terraform Backend Types

NEW QUESTION: 65
Which method for sharing Terraform modules fulfills the following criteria:
Keeps the module configurations confidential within your organization.
Supports Terraform's semantic version constraints.
Provides a browsable directory of your modules.
A. A Git repository containing your modules.
B. Public Terraform module registry.
C. A subfolder within your workspace.
D. HCP Terraform/Terraform Cloud private registry.
Answer: D (LEAVE A REPLY)
Confidentiality: UsingHCP Terraform/Terraform Cloud's private registrykeeps the module configurations within your organization, ensuring privacy and access control.
Version Constraints: The private registry supportssemantic versioning, allowing you to manage versions of your modules as Terraform does natively.
Browsable Directory: The private registry offers a user interface to browse modules, making it easy for users within the organization to locate and manage modules.
This setup aligns with HashiCorp's design forprivate registry supportin Terraform, meeting all listed requirements for secure, version-controlled, and searchable module storage.

NEW QUESTION: 66
Where can Terraform not load a provider from?
A. Plugins directory
B. Provider plugin chance
C. Official HashCrop Distribution on releases.hashcrop.com
D. Source code

Answer: D (LEAVE A REPLY)
This is where Terraform cannot load a provider from, as it requires a compiled binary file that implements the provider protocol. You can load a provider from a plugins directory, a provider plugin cache, or the official HashiCorp distribution on releases.hashicorp.com.

NEW QUESTION: 67
The public Terraform Module Registry is free to use.
A. True
B. False
Answer: (SHOW ANSWER)
The public Terraform Module Registry is free to use, as it is a public service that hosts thousands of self-contained packages called modules that are used to provision infrastructure. You can browse, use, and publish modules to the registry without any cost.

NEW QUESTION: 68
You're writing a Terraform configuration that needs to read input from a local file called id_rsa.pub. Which built-in Terraform function can you use to import the file's contents as a string?
A. fileset("id_rsa.pub")
B. file("id_rsa.pub")
C. filebase64("id_rsa.pub")
D. templatefile("id_rsa.pub")
Answer: B (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (B):The file() function reads a file from disk and returns its content as a string. This is commonly used for public keys (.pub files).
* Analysis of Incorrect Options:
* A. fileset(): Returns a set of filenames matching a pattern, not file contents.
* C. filebase64(): Reads file and returns Base64 encoded string - unnecessary for .pub.
* D. templatefile(): Used for rendering templates with variables, not raw file contents.
* Key Concept:Terraform's file() function is used for injecting file content directly.
Reference:Terraform Exam Objective - Read, Generate, and Modify Configurations.

NEW QUESTION: 69
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
Answer: B (LEAVE A REPLY)

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.

NEW QUESTION: 70
Which of the following is availableonlyinHCP Terraform workspacesandnot in Terraform CLI?
A. Dry runs with terraform plan.
B. Secure variable storage.
C. Using one workspace's state as a data source for another.
D. Support for multiple cloud providers.
Answer: (SHOW ANSWER)
C (#Correct)- InHCP Terraform, you can use the terraform_remote_state data source to referenceanother workspace's state. This feature isnot available in Terraform CLI.
A (#Incorrect)- terraform plan (dry runs)is available in both Terraform CLI and HCP Terraform.
B (#Incorrect)- Secure variable storageexists in HCP Terraform, but CLI users can store variables securely using environment variables or .tfvars files.
D (#Incorrect)- BothTerraform CLI and HCP Terraform support multiple cloud providers.
Official Terraform Documentation Reference:
terraform_remote_state - HashiCorp Documentation

NEW QUESTION: 71
When you run terraform apply, the Terraform CLI will print output values from both the root module and any child modules.
A. True
B. False
Answer: A (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (True):When terraform apply completes successfully, Terraform prints output values. Outputs from both root and child modules are displayed, but child module outputs must be explicitly exposed through the root module outputs to be visible at the CLI.
* Analysis of Incorrect Option:
* False: Incorrect, because Terraform does display output values, but only if they are exposed from child modules to the root module.
* Key Concept:Outputs help you extract important information (e.g., IP addresses, resource IDs) from your configuration.
Reference:Terraform Exam Objective - Read, Generate, and Modify Configurations.

NEW QUESTION: 72
Which of the following is not a valid source path for specifying a module?
A. source - "github.com/hashicorp/examplePref-ul.0.8M
B. source = "./module?version=vl.6.0"
C. source - "hashicorp/consul/aws"

D. source - "./module"

Answer: B (LEAVE A REPLY)

Terraform modules are referenced by specifying a source location. This location can be a URL or a file path.

However, specifying query parameters such as?version=vl.6.0directly within the source path is not a valid or supported method for specifying a module version in Terraform. Instead, version constraints are specified using theversionargument within the module block, not as part of the source string.References= This clarification is based on Terraform's official documentation regarding module usage, which outlines the correct methods for specifying module sources and versions.

NEW QUESTION: 73

In Terraform HCL, an object type of object({name=string, age-number}) would match this value.

A.
```
{
  name = "John"
  age = fifty two
}
```

HashiCorp

B.
```
{
  name = "John"
  age = 52
}
```

C.
```
{
  name =  John
  age = fifty two
}
```

D.
```
{
  name =  John
  age = 52
}
```


A. Option A
B. Option B
C. Option C
D. Option D
Answer: B (LEAVE A REPLY)
From the official Terraform Type Constraints Documentation:
The object({ name = string, age = number }) type expects:
name to be aquoted string, e.g. "John"

age to be anumber, e.g. 52
Option Bsatisfies both:
{
name = "John" ##Valid string
age = 52 ##Valid number
}
Let's analyze the incorrect ones:
#Option A: "fifty two" is not a valid number.
#Option C: John is unquoted (invalid string), and "fifty two" is not a number.
#Option D: name = John is not quoted, making it an invalid string in HCL.
Terraform is strict about type and formatting. Stringsmust be quoted, and numbersmust be numeric literals.

NEW QUESTION: 74
What is modified when executing Terraform inrefresh-only mode?
A. Your Terraform configuration.
B. Your Terraform plan.
C. Your state file.
D. Your cloud infrastructure.
Answer: C (LEAVE A REPLY)
Inrefresh-only mode(terraform apply -refresh-only), Terraformupdates the state fileto match the actual infrastructurewithout modifying resources.
A (Terraform configuration)-Not modifiedin refresh-only mode.
B (Terraform plan)-Plan is generated but not modified.
D (Cloud infrastructure)-Not modifiedin refresh-only mode.
Official Terraform Documentation Reference:
Refresh-Only Mode

NEW QUESTION: 75
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
Answer: (SHOW ANSWER)
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 any terraform plan or terraform apply operation. You can write aSentinel 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. References = [Sentinel Policy Framework], [Manage Policies in Terraform Cloud], [Write and Test Sentinel Policies for Terraform]

NEW QUESTION: 76
In a Terraform Cloud workpace linked to a version control repository speculative plan rum start automatically commit changes to version control.
A. True
B. False
Answer: (SHOW ANSWER)
When you use a remote backend that needs authentication, HashiCorp recommends that you:

NEW QUESTION: 77
Before you can use a new backend or HCP Terraform/Terraform Cloud integration, you must first execute terraform init.
A. True
B. False
Answer: A (LEAVE A REPLY)
The terraform init command is required before using a new backend or integrating with HCP Terraform
/Terraform Cloud.
* terraform init initializes the working directory and sets up the backend, downloading necessary provider plugins and modules.
* If the backend configuration changes, Terraform willrequirere-initialization to apply those changes.
* Without running terraform init, Terraform willfailto use a new backend or remote Terraform Cloud workspace.

Official Terraform Documentation Reference:
terraform init - HashiCorp Documentation

NEW QUESTION: 78
Which command(s) adds existing resources in a public cloud into Terraform state?
A. terraform init
B. terraform plan
C. terraform refresh
D. terraform import
E. All of these
Answer: D (LEAVE A REPLY)
Importing Existing Resources: The terraform import command brings resources already deployed in a cloud environment into Terraform's state file, allowing Terraform to manage them.
Workflow Usage: Importing is vital when managing resources created outside of Terraform or those in place before Terraform adoption.
Refer to Terraform's import command documentation.

NEW QUESTION: 79
Terraformrequiresthe Go runtime as a prerequisite for installation.
A. True
B. False
Answer: B (LEAVE A REPLY)
Terraformis written in Go, but it isdistributed as a standalone binaryanddoes notrequire the Go runtime.
Users do not need to install Go to run Terraform.
Official Terraform Documentation Reference:
Terraform Install Guide

NEW QUESTION: 80
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
Answer: A (LEAVE A REPLY)
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.

NEW QUESTION: 81
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
Answer: B (LEAVE A REPLY)
The .terraform.lock.hcl file 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.

NEW QUESTION: 82
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
Answer: A (LEAVE A REPLY)
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. References = [Command: fmt]2

NEW QUESTION: 83
Parent modules can always access a child module's variable values.
A. True
B. False
Answer: B (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct answer:Parent modules cannot directly access child module variables. Only values explicitly exposed via outputs can be accessed by the parent module.
* Analysis of Incorrect Options (Distractors):

* A. True: Incorrect because module variable scope is isolated.
* Key Concept:Terraform enforces module encapsulation, exposing data only through outputs.
Reference:Terraform Exam Objective - Interact with Terraform Modules

NEW QUESTION: 84
terraform validate uses provider APIs to verify your infrastructure settings.
A. True
B. False
Answer: B (LEAVE A REPLY)
terraform validateonlychecks the configuration's syntax and internal consistency-itdoes notinteract with provider APIs or check if the infrastructure settings are correct.
It ensures that the Terraform code is syntactically correct and follows proper HCL structure.
However, it doesnotverify if resources are valid according to the provider API or if the credentials are correct.
To verify actual infrastructure settings, use terraform plan, which interacts with the provider APIs.
Official Terraform Documentation Reference:
terraform validate - HashiCorp Documentation

NEW QUESTION: 85
You have used Terraform lo create an ephemeral development environment in the (loud and are now ready to destroy all the Infrastructure described by your Terraform configuration To be safe, you would like to first see all the infrastructure that Terraform will delete.
Which command should you use to show all of the resources that mil be deleted? Choose two correct answers.
A. Run terraform state rm '
B. Run terraform show :destroy
C. Run terraform destroy and it will first output all the resource that will be deleted before prompting for approval
D. Run terraform plan .destory
Answer: C,D (LEAVE A REPLY)
To see all the resources that Terraform will delete, you can use either of these two commands:
terraform destroy will show the plan of destruction and ask for your confirmation before proceeding. You can cancel the command if you do not want to destroy the resources.
terraform plan -destroy will show the plan of destruction without asking for confirmation. You can use this command to review the changes before running terraform destroy. References = :
Destroy Infrastructure : Plan Command: Options

NEW QUESTION: 86
Which of the following is true about terraform apply?(Pick 2 correct responses)
A. You must pass the output of a terraform plan command to it.
B. By default, it does not refresh your state file to reflect the current infrastructure configuration.

C. Depending on provider specification, Terraform may need to destroy and recreate your infrastructure resources.
D. You cannot target specific resources for the operation.
E. It only operates on infrastructure defined in the current working directory or workspace.
Answer: (SHOW ANSWER)
* C (#Correct)- If changes require aresource replacement(e.g., changing an immutable attribute like instance type), Terraform willdestroy and recreate the resource.
* E (#Correct)- Terraform only appliesthe configuration in the current directory or workspace.

NEW QUESTION: 87
You have created a main.tf Terraform configuration consisting of an application server, a database and a load balanced. You ran terraform apply and Terraform created all of the resources successfully.
Now you realize that you do not actually need the load balancer, so you run terraform destroy without any flags. What will happen?
A. Terraform will prompt you to pick which resource you want to destroy
B. Terraform will destroy the application server because it is listed first in the code
C. Terraform will prompt you to confirm that you want to destroy all the infrastructure
D. Terraform will destroy the main, tf file
E. Terraform will immediately destroy all the infrastructure
Answer: C (LEAVE A REPLY)
This is what will happen if you run terraform destroy without any flags, as it will attempt to delete all the resources that are associated with your current working directory or workspace. You can use the -target flag to specify a particular resource that you want to destroy.

NEW QUESTION: 88
Which of the following methods, used to provision resources into a public cloud, demonstrates the concept of infrastructure as code?
A. curl commands manually run from a terminal
B. A sequence of REST requests you pass to a public cloud API endpoint Most Voted
C. A script that contains a series of public cloud CLI commands
D. A series of commands you enter into a public cloud console
Answer: C (LEAVE A REPLY)
The concept of infrastructure as code (IaC) is to define and manage infrastructure using code, rather than manual processes or GUI tools. A script that contains a series of public cloud CLI commands is an example of IaC, because it uses code to provision resources into a public cloud. The other options are not examples of IaC, because they involve manual or interactive actions, such as running curl commands, sending REST requests, or entering commands into a console. References = [Introduction to Infrastructure as Code with Terraform] and [Infrastructure as Code]

NEW QUESTION: 89

Which of these are benefits of using Sentinel with HCP Terraform/Terraform Cloud? (Pick the 3 correct responses)
A. You can enforce a list of approved AWS AMIs.
B. Sentinel Policies can be written in HashiCorp Configuration Language (HCL).
C. You can check out and check in cloud access keys.
D. Policy-as-code can enforce security best practices.
Answer: (SHOW ANSWER)
Sentinel is a policy-as-code framework that integrates with Terraform Cloud to enforce security, compliance, and governance rules. You can enforce rules such as approved AMIs and ensure security best practices.
Policies are written in the Sentinel language, not HCL.
References:
* Sentinel Policies

NEW QUESTION: 90
What task does the terraform import command perform?
A. Imports resources from one Terraform state file to another.
B. Imports existing resources into Terraform's state file.
C. Imports a new Terraform module into Terraform's state file.
D. Imports all infrastructure from the configured cloud provider.
E. Imports provider configuration from one state file to another.
Answer: (SHOW ANSWER)
Detailed Explanation:
* Rationale for Correct Answer (B):terraform import allows Terraform to associate existing resources (created outside of Terraform or by another tool) with a Terraform configuration by writing them into the state file. After import, Terraform can manage those resources.
* Analysis of Incorrect Options:
* A. From one state file to another: Incorrect, import does not transfer between state files.
* C. Importing a module: Incorrect, modules are defined in configuration, not imported.
* D. Import all infrastructure: Incorrect, import is per-resource, not bulk.
* E. Provider configuration transfer: Incorrect, provider configs are in .tf files, not imported with this command.
* Key Concept:The terraform import command bridges existing resources with Terraform state management.
Reference:Terraform Exam Objective - Implement and Maintain State.

NEW QUESTION: 91
You have developed a new cloud-based service that uses proprietary APIs and want to use Terraform to create, manage, and delete users from the service. How can Terraform interact with the service?
A. Terraform can manage users for any service that is hosted on a public cloud provider.

B. Develop and publish a custom provider to interact with the service using its proprietary APIs.
Answer: B (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (B):Terraform interacts with external systems through providers. Since this is a new proprietary service with its own API, Terraform does not support it by default. You would need to develop a custom provider that implements Terraform's provider interface and interacts with the proprietary APIs to manage resources like users.
* Analysis of Incorrect Option:
* A. Any service hosted on a public cloud provider: Not true. Terraform can only manage services that have providers (official or custom). Just being "hosted on a public cloud" doesn't guarantee Terraform support.
* Key Concept:Terraform providers act as the bridge between Terraform and external APIs. For unsupported services, a custom provider must be developed.
Reference:Terraform Exam Objective - Manage Terraform Resources and Providers.

NEW QUESTION: 92
If a module declares a variable with a default, that variable must also be defined within the module.
A. True
B. False
Answer: B (LEAVE A REPLY)
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. References =
[Module Variables]

NEW QUESTION: 93
A senior admin accidentally deleted some of your cloud instances. What will Terraform do when you run terraform apply?
A. Tear down the entire workspace's infrastructure and rebuild it.
B. Build a completely brand new set of infrastructure.

C. Rebuild only the instances that were deleted.
D. Stop and generate an error message about the missing instances.
Answer: C (LEAVE A REPLY)
Terraform detects infrastructure drift by comparing thestate filewith the actual infrastructure.
* When an instance ismanually deleted, Terraformsees it as missingand marks it for recreation.
* Running terraform apply willonly recreate the missing instanceswhile leaving the rest of the infrastructure unchanged.
Explanation of incorrect answers:
* A (Tear down everything and rebuild)- Incorrect. Terraform does not destroy existing infrastructure unless explicitly told to.
* B (Build a completely new set of infrastructure)- Incorrect. Terraform does not create duplicates unless configuration changes.
* D (Stop and error out)- Incorrect. Terraform does not fail; itrebuilds missing resourcesautomatically.
Official Terraform Documentation Reference:
Handling Infrastructure Drift

NEW QUESTION: 94
Which of the following is not a way to trigger terraform destroy?
A. Using the destroy command with auto-approve.
B. Passing --destroy at the end of a plan request.
C. Running terraform destroy from the correct directory and then typing yes when prompted in the CLI.
Answer: (SHOW ANSWER)
Destroy Command Options: The terraform destroy command is the correct method to destroy resources, and it requires either manual approval or the -auto-approve flag.
Unsupported Triggering Method: The --destroy flag is not a recognized option for plan or apply commands, making B the correct answer as it is not a valid way to initiate resource destruction.
For more on destroying resources, refer to the terraform destroy command documentation in the Terraform CLI reference.

NEW QUESTION: 95
Which of these are features of HCP Terraform/Terraform Cloud? Pick the 2 correct responses below.
A. Automated infrastructure deployment visualization.
B. A web-based user interface (UI).
C. Automatic backups of configuration and state.
D. Remote state storage.
Answer: A,D (LEAVE A REPLY)
* Automated Visualization: HCP Terraform providesvisualization toolsthat map infrastructure configurations, helping users manage complex architectures.

* Remote State Storage: Terraform Cloud offersremote state management, essential for teams working collaboratively on shared infrastructure, ensuring consistency and avoiding state conflicts.
For more information, consultTerraform Cloud and HCP Terraform featuresin the official documentation.

NEW QUESTION: 96
Terraform variable names are saved in the state file.
A. True
B. False
Answer: (SHOW ANSWER)
Terraform variable names are not saved in the state file, only their values are. The state file only stores the attributes of the resources and data sources that are managed by Terraform, not the variables that are used to configure them.

NEW QUESTION: 97
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
Answer: D (LEAVE A REPLY)
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.

NEW QUESTION: 98
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]
Answer: (SHOW ANSWER)
To tell Terraform to stop managing a specific resource without destroying it, you can use the terraform state rm command. 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 use terraform import to start managing it again in a different configuration or workspace. The syntax for this command is terraform state rm <address>, where <address> is 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 the aws_instance resource named ubuntu from the state.
References: Command: state rm : Moving Resources

NEW QUESTION: 99
Which are forbidden actions when the terraform state file is locked? Choose three correct answers.
A. Terraform state list
B. Terraform destroy
C. Terraform validate
D. Terraform validate
E. Terraform for
F. Terraform apply
Answer: B,C,F (LEAVE A REPLY)
The terraform state file is locked when a Terraform operation that could write state is in progress. This prevents concurrent state operations that could corrupt the state. The forbidden actions when the state file is locked are those that could write state, such as terraform apply, terraform destroy, terraform refresh, terraform taint, terraform untaint, terraform import, and terraform state *. The terraform validate command is also forbidden, because it requires an initialized working directory with the state file. The allowed actions when the state file is locked are those that only read state, such as terraform plan, terraform show, terraform output, and terraform console.
References = [State Locking] and [Command: validate]

NEW QUESTION: 100
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
Answer: E (LEAVE A REPLY)
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.

NEW QUESTION: 101

While attempting to deploy resources into your cloud provider using Terraform, you begin to see some odd behavior and experience slow responses. In order to troubleshoot you decide to turn on Terraform debugging.
Which environment variables must be configured to make Terraform's logging more verbose?
A. TF_LOG_PAIH
B. TF_LOG
C. TF_VAR_log_path
D. TF_VAR_log_level
Answer: B (LEAVE A REPLY)
To make Terraform's logging more verbose for troubleshooting purposes, you must configure theTF_LOG environment variable. This variable controls the level of logging and can be set to TRACE, DEBUG, INFO, WARN, or ERROR, with TRACE providing the most verbose output.References= Detailed debugging instructions and the use of environment variables likeTF_LOGfor increasing verbosity are part of Terraform's standard debugging practices

NEW QUESTION: 102
A resource block is shown in the Exhibit section of this page. How would you reference the attribute name of this resource in HCL?
A. resource.kubernetes_namespace.example.name
B. kubernetes_namespace.example.name
C. data.kubernetes.namespace.name
D. kubernetes_namespace.test.name
Answer: (SHOW ANSWER)
In Terraform, the correct way to reference a resource attribute is:
pgsql
CopyEdit
resource_type.resource_name.attribute
For example, if the resource block is:
hcl
CopyEdit
resource "kubernetes_namespace" "example" {
metadata {
name = "my-namespace"
}
}
To reference the name attribute, use:
pgsql
CopyEdit
kubernetes_namespace.example.name
Explanation of incorrect answers:

A (resource.kubernetes_namespace.example.name)- Incorrect. Terraform does not use the resource. prefix when referencing resources.
C (data.kubernetes.namespace.name)- Incorrect. This syntax is used fordata sources, not resources.
D (kubernetes_namespace.test.name)- Incorrect. The resource name is "example", not "test".
Official Terraform Documentation Reference:
Terraform Resource References

NEW QUESTION: 103
Why does this backend configuration not follow best practices?

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
Answer: C (LEAVE A REPLY)
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.

NEW QUESTION: 104
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: (SHOW ANSWER)

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. References = [Importing Infrastructure]

NEW QUESTION: 105

Which command add existing resources into Terraform state?

A. Terraform init

B. Terraform plan

C. Terraform refresh

D. Terraform import

E. All of these

Answer: D (LEAVE A REPLY)

This is the command that can add existing resources into Terraform state, by matching them with the corresponding configuration blocks in your files.

NEW QUESTION: 106

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

Answer: C (LEAVE A REPLY)

This is the best time to run terraform 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.

Valid Terraform-Associate-003 Dumps shared by Actual4test.com for Helping Passing Terraform-Associate-003 Exam! Actual4test.com now offer the **newest Terraform-Associate-003 exam dumps**, the Actual4test.com Terraform-Associate-003 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com Terraform-Associate-003 dumps with Test Engine here: https://www.actual4test.com/Terraform-Associate-003_examcollection.html (**256** Q&As Dumps, **30%OFF** **Special Discount:** **Freepdfdumps**)