HashiCorp.Terraform-Associate-003.v2025-11-15.q123

Exam Code:	Terraform-Associate-003
Exam Name:	HashiCorp Certified: Terraform Associate (003) (HCTA0-003)
Certification Provider:	HashiCorp
Free Question Number:	123
Version:	v2025-11-15
# of views:	108
# of Questions views:	1230
https://www.freepdfdumps.com/HashiCorp.Terraform-Associate-003.v2025-11-15.q123.html	

NEW QUESTION: 1
Multiple team members are collaborating on infrastructure using Terraform and want to format the* Terraform code following standard Terraform-style convention.
How should they ensure the code satisfies conventions?
A. Terraform automatically formats configuration on terraform apply
B. Run terraform validate prior to executing terraform plan or terraform apply
C. Use terraform fmt
D. Replace all tabs with spaces
Answer: (SHOW ANSWER)
The terraform fmt command is used to format Terraform configuration files to a canonical format and style.
This ensures that all team members are using a consistent style, making the code easier to read and maintain.
It automatically applies Terraform's standard formatting conventions to your configuration files, helping maintain consistency across the team's codebase.
References:
Terraform documentation on terraform fmt: Terraform Fmt

NEW QUESTION: 2
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: 3
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: 4
Which statement describes a goal of Infrastructure as Code (IaC)?
A. A pipeline process to test and deliver software.
B. Write once, run anywhere.
C. The programmatic configuration of resources.
D. Defining a vendor-agnostic API.
Answer: C (LEAVE A REPLY)
Infrastructure as Code (IaC) refers to managing infrastructureprogrammatically, enabling automation and repeatability.
Ais incorrect becauseIaC is not just about software delivery pipelines; it specifically deals with infrastructure.
Bis incorrect because"Write once, run anywhere" applies more to software development than IaC.
Dis incorrect becauseIaC does not enforce vendor-agnostic APIs; it depends on the tool and provider.
Official Terraform Documentation Reference:
What is Infrastructure as Code?

NEW QUESTION: 5
Which of the following are advantages of using infrastructure as code (IaC) instead of provisioning with a graphical user interface (GUI)? Choose two correct answers.
A. Prevents manual modifications to your resources
B. Lets you version, reuse, and share infrastructure configuration
C. Secures your credentials
D. Provisions the same resources at a lower cost

E. Reduces risk of operator error

Answer: B,E (LEAVE A REPLY)

Infrastructure as code (IaC) is a way of managing and provisioning cloud infrastructure using programming techniques instead of manual processes1. IaC has many advantages over using a graphical user interface (GUI) for provisioning infrastructure, such as:

*Versioning: IaC allows you to store your infrastructure configuration in a version control system, such as Git, and track changes over time. This enables you to roll back to previous versions, compare differences, and collaborate with other developers2.

*Reusability: IaC allows you to create reusable modules and templates that can be applied to different environments, such as development, testing, and production. This reduces duplication, improves consistency, and speeds up deployment3.

*Sharing: IaC allows you to share your infrastructure configuration with other developers, teams, or organizations, and leverage existing code from open source repositories or registries. This fosters best practices, innovation, and standardization4.

*Risk reduction: IaC reduces the risk of human error, configuration drift, and security breaches that can occur when provisioning infrastructure manually or using a GUI. IaC also enables you to perform automated testing, validation, and compliance checks on your infrastructure before deploying it5.

References =

*1: What is Infrastructure as Code? Explained for Beginners - freeCodeCamp.org

*2: The benefits of Infrastructure as Code - Microsoft Community Hub

*3: Infrastructure as Code : Best Practices, Benefits & Examples - Spacelift

*4: 5 Benefits of Infrastructure as Code (IaC) for Modern Businesses in the Cloud

*5: The 7 Biggest Benefits of Infrastructure as Code - DuploCloud

NEW QUESTION: 6

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: 7

Your root module contains a variable namednum_servers. Which is the correct way to pass its value to a child module with an input namedservers?

A. servers = num_servers

B. servers = var(num_servers)

C. servers = var.num_servers
D. servers = ${var.num_servers}
Answer: C (LEAVE A REPLY)
The correct syntax to pass a variable from the root module to a child module isservers = var.num_servers.
Terraform uses dot notation to reference variables.
References:
Terraform Variables

NEW QUESTION: 8
What is the name of the default file where Terraform stores the state?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.
Answer:
Terraform.tfstate
Explanation:
The name of the default file where Terraform stores the state is terraform.tfstate. This file contains a JSON representation of the current state of the infrastructure managed by Terraform. Terraform uses this file to track the metadata and attributes of the resources, and to plan and apply changes. By default, Terraform stores the state file locally in the same directory as the configuration files, but it can also be configured to store the state remotely in a backend.
References = [Terraform State], [State File Format]

NEW QUESTION: 9
Which of these are features of Terraform Cloud? Choose two correct answers.
A. A web-based user interface (UI)
B. Automated infrastructure deployment visualization
C. Automatic backups
D. Remote state storage
Answer: A,D (LEAVE A REPLY)
Terraform Cloud includes several features designed to enhance collaboration and infrastructure management.
Two of these features are:
A web-based user interface (UI): This allows users to interact with Terraform Cloud through a browser, providing a centralized interface for managing Terraform configurations, state files, and workspaces.
Remote state storage: This feature enables users to store their Terraform state files remotely in Terraform Cloud, ensuring that state is safely backed up and can be accessed by team members as needed.

NEW QUESTION: 10

You cannot install third party plugins using terraform init.

A. True

B. False

Answer: (SHOW ANSWER)

You can install third party plugins using terraform init, as long as you specify the plugin directory in your configuration or as a command-line argument. You can also use the terraform providers mirror command to create a local mirror of providers from any source.

NEW QUESTION: 11

Which of these ate secure options for storing secrets for connecting to a Terraform remote backend? Choose two correct answers.

A. A variable file

B. Defined in Environment variables

C. Inside the backend block within the Terraform configuration

D. Defined in a connection configuration outside of Terraform

Answer: (SHOW ANSWER)

Environment variables and connection configurations outside of Terraform are secure options for storing secrets for connecting to a Terraform remote backend. Environment variables can be used to set values for input variables that contain secrets, such as backend access keys or tokens. Terraform will read environment variables that start with TF_VAR_ and match the name of an input variable. For example, if you have an input variable called backend_token, you can set its value with the environment variable TF_VAR_backend_token1. Connection configurations outside of Terraform are files or scripts that provide credentials or other information for Terraform to connect to a remote backend. For example, you can use a credentials file for the S3 backend2, or a shell script for the HTTP backend3. These files or scripts are not part of the Terraform configuration and can be stored securely in a separate location. The other options are not secure for storing secrets. A variable file is a file that contains values for input variables. Variable files are usually stored in the same directory as the Terraform configuration or in a version control system. This exposes the secrets to anyone who can access the files or the repository. You should not store secrets in variable files1. Inside the backend block within the Terraform configuration is where you specify the type and settings of the remote backend. The backend block is part of the Terraform configuration and is usually stored in a version control system. This exposes the secrets to anyone who can access the configuration or the repository. You should not store secrets in the backend block4. References = [Terraform Input Variables]1, [Backend Type: s3]2, [Backend Type: http]3, [Backend Configuration]4

NEW QUESTION: 12

How is terraform import run?

A. As a part of terraform init

B. As a part of terraform plan

C. As a part of terraform refresh

D. By an explicit call
E. All of the above
Answer: D (LEAVE A REPLY)
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: 13
The_____determines how Terraform creates, updates, or delete resources.
A. Terraform configuration
B. Terraform provisioner
C. Terraform provider
D. Terraform core
Answer: C (LEAVE A REPLY)
This is what determines how Terraform creates, updates, or deletes resources, as it is responsible for understanding API interactions with some service and exposing resources and data sources based on that API.

NEW QUESTION: 14
What feature stops multiple users from operating on the Terraform state at the same time?
A. State locking
B. Version control
C. Provider constraints
D. Remote backends
Answer: A (LEAVE A REPLY)
State locking prevents other users from modifying the state file while a Terraform operation is in progress. This prevents conflicts and data loss1.

NEW QUESTION: 15
You much initialize your working directory before running terraform validate.
A. True
B. False
Answer: (SHOW ANSWER)
You must initialize your working directory before running terraform validate, as it will ensure that all the required plugins and modules are installed and configured properly. If you skip this step, you may encounter errors or inconsistencies when validating your configuration files.

NEW QUESTION: 16
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: 17
A Terraform backend determines how Terraform loads state and stores updates when you execute which command?
A. apply
B. destroy
C. Both of these are correct.
D. Neither of these are correct.
Answer: (SHOW ANSWER)
A Terraformbackenddetermineswhere and how Terraform state is stored.
Bothterraform applyandterraform destroyinteract with state, so Terraform needs access to the backend for state storage and updates.
D (Neither are correct)is incorrect becauseTerraform state is required for both apply and destroy operations.
Official Terraform Documentation Reference:
Terraform Backends

NEW QUESTION: 18
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: 19
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.

NEW QUESTION: 20
What does Terraform not reference when running a terraform apply -refresh-only ?
A. State file
B. Credentials
C. Cloud provider
D. Terraform resource definitions in configuration files
Answer: (SHOW ANSWER)
When running a terraform apply -refresh-only, Terraform does not reference the configuration files, but only the state file, credentials, and cloud provider. The purpose of this command is to update the state file with the current status of the real resources, without making any changes to them1.

NEW QUESTION: 21
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: 22
terraform init retrieves and caches the configuration for all remote modules.
A. True
B. False
Answer: (SHOW ANSWER)
terraform initretrieves and caches providers and modules, butonly for explicitly declared modulesin the configuration.
If a module isadded or updated, terraform init needs to be re-run to download the new version.
Terraform does not automatically cache all remote modules unless they are explicitly referenced in the configuration.
Official Terraform Documentation Reference:
terraform init - HashiCorp Documentation

NEW QUESTION: 23
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: B (LEAVE A REPLY)

Destroy Command Options: Theterraform destroycommand is the correct method to destroy resources, and it requires either manual approval or the -auto-approve flag.

Unsupported Triggering Method: The--destroy flagis not a recognized option for plan or apply commands, making B the correct answer as it isnota valid way to initiate resource destruction.

For more on destroying resources, refer to theterraform destroy command documentationin the Terraform CLI reference.

NEW QUESTION: 24

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

A. True

B. False

Answer: B (LEAVE A REPLY)

You cannot reference a resource created with for_each using a splat (*) expression, as it will not work with resources that have non-numeric keys. You need to use a for expression instead to iterate over the resource instances.

NEW QUESTION: 25

If a module declares a variable without a default value, you must pass the value of the variable within the module block when you call the module in your configuration.

A. True

B. False

Answer: A (LEAVE A REPLY)

Detailed Explanation:

* Rationale for Correct Answer (True):Variables without defaults are required inputs. If the calling module doesn't supply a value, Terraform will fail with an error at plan time.

* Analysis of Incorrect Option:

* False: Incorrect, because Terraform does not assume defaults when none are provided.

* Key Concept:Terraform modules enforce required vs. optional variables depending on whether a default is set.

Reference:Terraform Exam Objective - Interact with Terraform Modules.

NEW QUESTION: 26

Terraform providers are part of the Terraform core binary.

A. True

B. False

Answer: B (LEAVE A REPLY)

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. References = : Providers - Configuration Language | Terraform : Terraform Registry - Providers Overview | Terraform

NEW QUESTION: 27
Which provider authentication method prevents credentials from being stored in the state file?
A. Using environment variables
B. Specifying the login credentials in the provider block
C. Setting credentials as Terraform variables
D. None of the above
Answer: (SHOW ANSWER)
None of the above methods prevent credentials from being stored in the state file. Terraform stores the provider configuration in the state file, which may include sensitive information such as credentials. This is a potential security risk and should be avoided if possible. To prevent credentials from being stored in the state file, you can use one of the following methods:
Use environment variables to pass credentials to the provider. This way, the credentials are not part of the provider configuration and are not stored in the state file. However, this method may not work for some providers that require credentials to be set in the provider block.
Use dynamic credentials to authenticate with your cloud provider. This way, Terraform Cloud or Enterprise will request temporary credentials from your cloud provider for each run and use them to provision your resources. The credentials are not stored in the state file and are revoked after the run is completed. This method is supported for AWS, Google Cloud Platform, Azure, and Vault. References = : [Sensitive Values in State] : Authenticate providers with dynamic credentials

NEW QUESTION: 28
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: 29
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 build a reusable code based that can deploy resources into any AWS region
B. The team is asked to manage a new application stack built on AWS-native services
C. The organization decides to expand into Azure wishes to deploy new infrastructure
D. The DevOps team is tasked with automating a manual, web console-based provisioning.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 30
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: 31
What is the best practice for securely managing sensitive values in Terraform?
A. In a plaintext document on a shared drive.
B. In a terraform.tfvars file, checked into version control.
C. In a terraform.tfvars file, securely managed and shared with your team.
D. In an HCP Terraform/Terraform Cloud variable, with the sensitive option checked.
E. In HashiCorp Vault.
Answer: C,D,E (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answers:
* C. terraform.tfvars securely managed: Acceptable if distributed securely outside version control.

* D. HCP Terraform/Terraform Cloud sensitive variables: Official best practice for team-based workflows.
* E. Vault: HashiCorp Vault is designed for secret management and integrates well with Terraform.
* Analysis of Incorrect Options:
* A: Storing plaintext secrets on shared drives is insecure.
* B: Checking secrets into version control is a major security risk.
* Key Concept:Sensitive values should be managed securely in Vault, HCP Terraform, or securely shared
.tfvars files - never in plaintext or version control.
Reference:Terraform Exam Objective - Use Terraform to Manage Infrastructure.

NEW QUESTION: 32
What type of information can be found on the Terraform Registry when using published modules?
A. Required input variables.
B. Outputs.
C. Optional input variables and default values.
D. All of the above.
Answer: D (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (D):The Terraform Registry provides documentation for published modules, including:
* Required inputs (variables you must supply).
* Optional inputs with defaults (so you know what you can override).
* Outputs (what values the module returns for use elsewhere).
Therefore, the correct answer is all of the above.
* Analysis of Incorrect Options:
* A. Required inputs only: Incomplete.
* B. Outputs only: Incomplete.
* C. Optional inputs only: Incomplete.
* D. All of the above: Correct because the Registry provides all relevant documentation.

* Key Concept:Terraform Registry modules include inputs, outputs, and usage details, enabling reusability and modular design.
Reference:Terraform Exam Objective - Interact with Terraform Modules.

NEW QUESTION: 33
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: 34
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: 35
When you use a backend that requires authentication, it is best practice to:
A. Run all of your Terraform commands on a shared server or container.
B. Configure the authentication credentials in your Terraform configuration files, and store them in a private version control system.
C. Use environment variables to configure authentication credentials outside of your Terraform configuration.
D. None of the above.
Answer: C (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (C):Best practice is to avoid hardcoding sensitive credentials in Terraform configurations or storing them in version control. Instead, credentials should be managed via environment variables, CLI authentication helpers, or secret managers (e.g., Vault).

* Analysis of Incorrect Options:
* A. Shared server: Not secure, introduces single point of failure and risk.
* B. Storing credentials in config files: A major security risk, especially if pushed to version control.
* D. None of the above: Incorrect, because option C is a valid and recommended approach.
* Key Concept:Security best practices in Terraform dictate that credentials should be externalized from Terraform code.
Reference:Terraform Exam Objective - Navigate Terraform State and Backends.

NEW QUESTION: 36
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: (SHOW ANSWER)
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: 37
Does Terraform enforce user-specific restrictions on plan files, preventing other users from applying them?
A. True
B. False
Answer: B (LEAVE A REPLY)

Detailed Explanation:
* Rationale for Correct Answer (False):Any user with access to the saved plan file (terraform plan - out=planfile) can run terraform apply planfile. Terraform does not enforce user-specific restrictions.
* Analysis of Incorrect Option:
* True: Incorrect - Terraform doesn't tie plan files to individual users.
* Key Concept:Plan files ensure predictability but are not bound to the identity of the user.
Reference:Terraform Exam Objective - Understand Terraform Basics and CLI.

NEW QUESTION: 38
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?
A. Data.vsphere_datacenter.DC.id
B. Vsphere_datacenter.dc.id
C. Data,dc,id
D. Data.vsphere_datacenter,dc
Answer: (SHOW ANSWER)
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 is data.
vsphere_datacenter.dc.id. This follows the syntax for accessing data source attributes, which is data.TYPE.
NAME.ATTRIBUTE. In this case, the data source type is vsphere_datacenter, the data source name is dc, and the attribute we want to access is id. The other options are incorrect becausethey either use the wrong syntax, the wrong punctuation, or the wrong case. References = [Data Source: vsphere_datacenter], [Data Source:
vsphere_folder], [Expressions: Data Source References]

NEW QUESTION: 39
Which method for sharing Terraform configurations fulfills the following criteria:
1. Keeps the configurations confidential within your organization
2. Support Terraform's semantic version constrains
3. Provides a browsable directory
A. Subfolder within a workspace
B. Generic git repository
C. Terraform Cloud private registry
D. Public Terraform module registry
Answer: C (LEAVE A REPLY)
This is the method for sharing Terraform configurations that fulfills the following criteria:
Keeps the configurations confidential within your organization
Supports Terraform's semantic version constraints

Provides a browsable directory
The Terraform Cloud private registry is a feature of Terraform Cloud that allows you to host and manage your own modules within your organization, and use them in your Terraform configurations with versioning and access control.

NEW QUESTION: 40
What is the workflow for deploying new infrastructure with Terraform?
A. Write Terraform configuration, run terraform init to initialize the working directory orworkspace, 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
Answer: (SHOW ANSWER)
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.

NEW QUESTION: 41
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: D (LEAVE A REPLY)
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: 42
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: (SHOW ANSWER)
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: 43
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: 44
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: 45
Which of these statements about HCP Terraform/Terraform Cloud workspaces is false?
A. They can securely store cloud credentials.
B. They have role-based access controls.
C. Plans and applies can be triggered via version control system integrations.
D. You must use the CLI to switch between workspaces.
Answer: D (LEAVE A REPLY)
In Terraform Cloud, you can switch between workspaces using both the web UI and CLI. The statement that you "must use the CLI" is false. Workspaces can securely store cloud credentials, offer role-based access control, and integrate with VCS to trigger plan and apply operations.
References:
Terraform Cloud Workspaces

NEW QUESTION: 46
Which of the following does terraform apply change after you approve the execution plan? (Choose two.)
A. Cloud infrastructure
B. The .terraform directory
C. The execution plan
D. State file
E. Terraform code
Answer: (SHOW ANSWER)
The terraform apply command 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 theconfiguration. It also updates the state file to reflect the new state of the infrastructure. The .
terraform directory, the execution plan, and the Terraform code are not changed by the terraform apply command. References = Command: apply and Purpose of Terraform State

NEW QUESTION: 47

_____backends support state locking.

A. All

B. No

C. Some

D. Only local

Answer: C (LEAVE A REPLY)

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.

NEW QUESTION: 48

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

Answer: C (LEAVE A REPLY)

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 -upgrade option.

NEW QUESTION: 49

What Terraform command always causes a state file to be updated with changes that might have been made outside of Terraform?

A. Terraform plan -refresh-only

B. Terraform show -json

C. Terraform apply -lock-false

D. Terraform plan target-state

Answer: (SHOW ANSWER)

This is the command that always causes a state file to be updated with changes that might have been made outside of Terraform, as it will only refresh the state file with the current status of the real resources, without making any changes to them or creating a plan.

NEW QUESTION: 50

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 run terraform init. This command initializes a Terraform working directory, which includes configuring the backend. If a remote

backend is specified, terraform init will 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 workingwith 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: 51
What command can you run to generateDOT (Graphviz)formatted data to visualize Terraform dependencies?
A. terraform refresh
B. terraform graph
C. terraform output
D. terraform show
Answer: B (LEAVE A REPLY)
terraform graphgeneratesGraphviz DOT formatoutput, which allows visualization ofresource dependenciesin Terraform.
A (terraform refresh)- Incorrect, as it only updates the state file.
C (terraform output)- Incorrect, as it only displays Terraform output values.
D (terraform show)- Incorrect, as it only displays the Terraform state.
Official Terraform Documentation Reference:
terraform graph - HashiCorp Documentation

NEW QUESTION: 52
You are writing a child Terraform module that provisions an AWS instance. You want to reference the IP address returned by the child module in the root configuration. You name the instance resource "main'.
Which of these is the correct way to define the output value?
A. Option A
B. Option B
C. Option C
D. Option D
Answer: A (LEAVE A REPLY)
According to theofficial Terraform documentationhere:
#Terraform Docs - Declaring an Output Value
In Terraform 0.12 and later, you can directly use expressions in outputs without interpolation syntax:
output "instance_ip_addr" {
value = aws_instance.server.private_ip
}
This aligns perfectly withOption A:
output "instance_ip_addr" {

value = aws_instance.main.private_ip
}
Why not the others?
#Option B: Incorrect syntax. "${main.private_ip}" is referencing main as if it were a global variable or resource, but it isn't defined. Plus, the output name is oddly written as aws_instance.instance_ip_addr, which is not a valid identifier format.
#Option C: Although valid in older versions (<= 0.11), interpolation with "${}" isdeprecatedin Terraform 0.12
+. The docs clearly advise using direct expressions instead.
#Option D: Terraform hasno return statement; this is syntactically invalid in Terraform's HCL.

NEW QUESTION: 53
Terraform configuration can only call modules from the public registry.
A. True
B. False
Answer: B (LEAVE A REPLY)
Terraform can call modules from various sources including the public Terraform Registry, private registries, local file paths, or version control systems like GitHub.
References:
Terraform Modules

NEW QUESTION: 54
By default, if you do not define a backend for your configuration, where does Terraform store information about the resources that it manages?
A. A subdirectory of your home directory named .terraform.d
B. A file in your configuration's directory named terraform.tfstate
C. A file in your configuration's directory named .terraform.lock.hcl
D. A subdirectory of your configuration named .terraform
Answer: B (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (B):Terraform uses a state file to map real-world resources to your configuration. By default, if no backend is explicitly defined, Terraform stores this information in a file named terraform.tfstate located in the same directory as your configuration files. This local state file is critical for Terraform to understand what infrastructure already exists and to plan updates correctly.
* Analysis of Incorrect Options (Distractors):
* A. .terraform.d (home directory subdirectory):This is used for storing plugins and some global settings, not for resource state.
* C. .terraform.lock.hcl (lock file):This file locks provider versions to ensure reproducible runs, but it does not contain resource state information.

* D. .terraform (subdirectory):This folder contains cached provider binaries and related data, but not the actual state file.
* Key Concept:Terraform state management is at the core of "Implement and Maintain State." Understanding where Terraform stores the state by default is critical because state files should often be stored remotely (using a backend like S3, GCS, or Terraform Cloud) for collaboration and reliability.
Reference:Terraform Exam Objective - Implement and Maintain State (HashiCorp Certified: Terraform Associate).

NEW QUESTION: 55
You created infrastructure outside the Terraform workflow that you now want to manage using Terraform.
Which command brings the infrastructure into Terraform state?
A. terraform get
B. terraform refresh
C. terraform import
D. terraform init
Answer: C (LEAVE A REPLY)
The terraform import command allows Terraform to takeexistinginfrastructure andbring it under Terraform's management.
A (terraform get)is incorrect because it is used to fetch modules.
B (terraform refresh)is incorrect because it only updates Terraform's state to match the infrastructure but does not import resources.
D (terraform init)is incorrect because it only initializes the Terraform working directory.
Official Terraform Documentation Reference:
terraform import - HashiCorp Documentation

NEW QUESTION: 56
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: 57

A developer accidentally launched a VM (virtual machine) outside of the Terraform workflow and ended up with two servers with the same name. They don't know which VM Terraform manages but do have a list of all active VM IDs.
Which of the following methods could you use to discover which instance Terraform manages?
A. Run terraform state list to find the names of all VMs, then run terraform state show for each of them to find which VM ID Terraform manages
B. Update the code to include outputs for the ID of all VMs, then run terraform plan to view the outputs
C. Run terraform taint/code on all the VMs to recreate them
D. Use terraform refresh/code to find out which IDs are already part of state
Answer: (SHOW ANSWER)
The terraform state list command lists all resources that are managed by Terraform in the current state file1. The terraform state show command shows the attributes of a single resource in the state file2. By using these two commands, you can compare the VM IDs in your list with the ones in the state file and identify which one is managed by Terraform.

NEW QUESTION: 58
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.6.0 directly within the source path is not a valid or supported method for specifying a module version in Terraform. Instead, version constraints are specified using the version argument 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: 59
You used Terraform to create an ephemeral development environment in the cloud and are now ready to destroy all the infrastructure described by your Terraform configuration. To be safe, you would like to first see all the infrastructure that Terraform will delete.
Which command should you use to show all the resources that will be deleted? (Pick the 2 correct responses)
A. Runterraform destroy. This will output all the resources that will be deleted before prompting for approval.
B. Runterraform show -destroy.

C. Runterraform state rm *.
Answer: A,B (LEAVE A REPLY)
Runningterraform destroywill show all resources that will be deleted before prompting for approval. You can also runterraform plan -destroyto simulate the destruction without actually applying it, which is useful for reviewing the planned changes.
References:
Terraform Destroy

NEW QUESTION: 60
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 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. References = : Command: fmt : Using Terraform fmt Command to Format Your Terraform Code

NEW QUESTION: 61
Your DevOps team is currently using the local backend for your Terraform configuration. You would like to move to a remote backend to store the state file in a central location. Which of the following backends would not work?
A. Artifactory
B. Amazon S3
C. Terraform Cloud
D. Git
Answer: (SHOW ANSWER)
This is not a valid backend for Terraform, as it does not support locking or versioning of state files4. The other options are valid backends that can store state files in a central location.

NEW QUESTION: 62
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 the TF_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 like TF_LOG for increasing verbosity are part of Terraform's standard debugging practices

NEW QUESTION: 63
terraform validate confirms thesyntaxof Terraform files.
A. True
B. False
Answer: A (LEAVE A REPLY)
terraform validatechecks for syntax errors and internal consistencyin the configuration files. However, itdoes notcheck if resource configurations are valid with the provider.
Official Terraform Documentation Reference:
terraform validate - HashiCorp Documentation

NEW QUESTION: 64
Which of these commands makes your code more human readable?
A. Terraform validate
B. Terraform output
C. Terraform show
D. Terraform file
Answer: (SHOW ANSWER)
The command that makes your code more human readable is terraform 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 of terraform fmt very important. You can run this command on your configuration files before committing them to source control or as part of your CI/CD pipeline. References = : Command: fmt : Using Terraform fmt Command to Format Your Terraform Code

NEW QUESTION: 65
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: (SHOW ANSWER)
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: 66
What value does the Terraform Cloud private registry provide over the public Terraform Module Registry?
A. The ability to share modules publicly with any user of Terraform
B. The ability to restrict modules to members of Terraform Cloud or Enterprise organizations
C. The ability to tag modules by version or release
D. The ability to share modules with public Terraform users and members of Terraform Cloud Organizations
Answer: B (LEAVE A REPLY)
The Terraform Cloud private registry provides the ability to restrict modules to members of Terraform Cloud or Enterprise organizations. This allows you to share modules within your organization without exposing them to the public. The private registry also supports importing modules from your private VCS repositories.
The public Terraform Module Registry, on the other hand, publishes modules from public Git repositories and makes them available to any user of Terraform. References = : Private Registry - Terraform Cloud : Terraform Registry - Provider Documentation

NEW QUESTION: 67
Terraform providers are always installed from the Internet.
A. True
B. False
Answer: B (LEAVE A REPLY)

Terraform providers are not always installed from the Internet. There are other ways to install provider plugins, such as from a local mirror or cache, from a local filesystem directory, or from a network filesystem.
These methods can be useful for offline or air-gapped environments, or for customizing the installation process. You can configure the provider installation methods using the provider_installation block in the CLI configuration file.

NEW QUESTION: 68
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: 69
What functionality do providers offer in Terraform?(Pick 3 correct responses)
A. Interact with cloud provider APIs.
B. Provision resources for on-premises infrastructure services.
C. Group a collection of Terraform configuration files that map to a single state file.
D. Provision resources for public cloud infrastructure services.
E. Enforce security and compliance policies.
Answer: A,B,D (LEAVE A REPLY)
A (#Correct)- Providers allow Terraform tointeract with APIsof cloud/on-premises services.
B (#Correct)- Some Terraform providers can provisionon-premises infrastructure, such as VMware, OpenStack, etc.
C (#Incorrect)- This describesTerraform Workspaces, not providers.
D (#Correct)- Terraform providers allow provisioning ofpublic cloud resources(AWS, Azure, GCP, etc.).
E (#Incorrect)- Enforcing security and compliance policies isnot a direct provider function, but it can be done using Sentinel or other policy-as-code tools.
Official Terraform Documentation Reference:
Terraform Providers

NEW QUESTION: 70
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: 71
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: 72
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: 73
Which command or set of commands will allow you to determine which virtual machine (VM) Terraform is managing?
A. Modify the Terraform configuration to add an import block for both of the virtual machines.
B. Run a terraform apply -refresh to identify the virtual machine IDs that are already managed by Terraform.
C. Run terraform state rm on both VMs, then terraform apply to recreate the correct one.
D. Run terraform state list to find the names of all VMs, then run terraform state show for each of them to find which VM ID Terraform manages.
Answer: (SHOW ANSWER)
Detailed Explanation:
* Rationale for Correct Answer (D):terraform state list shows all resources currently managed by Terraform. terraform state show <resource> provides detailed attributes, including the VM ID. This lets you match the Terraform-managed instance with the actual infrastructure.
* Analysis of Incorrect Options:
* A: Importing is not needed since one VM is already in state.
* B: terraform apply doesn't show which VM ID is managed, it only refreshes attributes.
* C: Removing state entries is destructive and may lead to losing state data unnecessarily.
* Key Concept:The state file is Terraform's source of truth for which resources it manages.
Reference:Terraform Exam Objective - Implement and Maintain State.

NEW QUESTION: 74
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: 75

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

Answer: (SHOW ANSWER)

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.

NEW QUESTION: 76

You can develop a custom provider to manage its resources using Terraform.

A. True

B. False

Answer: A (LEAVE A REPLY)

You can develop a custom provider to manage its resources using Terraform, as Terraform is an extensible tool that allows you to write your own plugins in Go language. You can also publish your custom provider to the Terraform Registry or use it privately.

NEW QUESTION: 77
When should you run terraform init?
A. Every time you run terraform apply
B. Before you start coding a new Terraform project
C. After you run terraform plan for the time in a new terraform project and before you run terraform apply
D. After you start coding a new terraform project and before you run terraform plan for the first time.
Answer: (SHOW ANSWER)
You should run terraform init after you start coding a new Terraform project and before you run terraform plan for the first time. This command will initialize the working directory by downloading the required providers and modules, creating the initial state file, and performing other necessary tasks. References = :
Initialize a Terraform Project

NEW QUESTION: 78
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. file("id_rsa.pub")
B. templaTefil("id_rsa.pub")
C. filebase64("id_rsa.pub")
D. fileset<"id_rsa.pub")
Answer: A (LEAVE A REPLY)
To import the contents of a local file as a string in Terraform, you can use the built-in file function. By specifying file("id_rsa.pub"), Terraform reads the contents of the id_rsa.pub file and uses it as a string within your Terraform configuration. This function is particularly useful for scenarios

where you need to include file data directly into your configuration, such as including an SSH public key for provisioning cloud instances.
References = This information is a standard part of Terraform's functionality with built-in functions, as outlined in Terraform's official documentation and commonly used in various Terraform configurations.

NEW QUESTION: 79
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: D (LEAVE A REPLY)
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 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. References = [Sentinel Policy Framework], [Manage Policies in Terraform Cloud], [Write and Test Sentinel Policies for Terraform]

NEW QUESTION: 80
What kind of configuration block will create an infrastructure object with settings specified within the block?
A. provider
B. state
C. data
D. resource
Answer: D (LEAVE A REPLY)
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.

NEW QUESTION: 81
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: 82
How do you specify a module's version when publishing it to the public terraform Module Registry?
A. Configuration it in the module's Terraform code
B. Mention it on the module's configuration page on the Terraform Module Registry
C. The Terraform Module Registry does not support versioning modules
D. Tag a release in the associated repo
Answer: (SHOW ANSWER)
This is how you specify a module's version when publishing it to the public Terraform Module Registry, as it uses the tags from your version control system (such as GitHub or GitLab) to identify module versions. You need to use semantic versioning for your tags, such as v1.0.0.

NEW QUESTION: 83
terraform plan updates your state file.
A. True
B. False
Answer: B (LEAVE A REPLY)
The terraform plan command does not update the state file. Instead, it reads the current state and the configuration files to determine what changes would be made to bring the real-world infrastructure into the desired state defined in the configuration. The plan operation is a read-only operation and does not modify the state or the infrastructure. It is the terraform apply command that actually applies changes and updates the state file.References = Terraform's official guidelines and documentation clarify the purpose of the terraform plan command, highlighting its role in preparing and showing an execution plan without making any changes to the actual state or infrastructure .

NEW QUESTION: 84
Which of these actions are forbidden when the Terraform state file is locked? (Pick the 3 correct responses)
A. terraform apply
B. terraform state list
C. terraform destroy
D. terraform fmt
Answer: (SHOW ANSWER)

When the state file is locked, operations that modify or depend on the state (liketerraform apply,terraform destroy, andterraform state list) are blocked.terraform fmtonly formats the configuration files and does not interact with the state, so it is allowed.
References:
Terraform State Locking

NEW QUESTION: 85
You must use different Terraform commands depending on the cloud provider you use.
A. True
B. False
Answer: B (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.
From Terraform CLI Overview:
"Terraform provides a consistent CLI workflow regardless of which cloud or service you're managing." Thesame Terraform commandswork across all providers via plugins.

NEW QUESTION: 86
terraform destroy is the only way to remove infrastructure.
A. True
B. False
Answer: (SHOW ANSWER)
While terraform destroy is a common way to remove infrastructure, it isnot the only way.
You can also remove resources bydeleting them from the configuration and running terraform apply.
Manually deleting resources in the cloud provider can also remove infrastructure (but Terraform won't be aware unless the state is updated).
Official Terraform Documentation Reference:
terraform destroy - HashiCorp Documentation

NEW QUESTION: 87
Which of the following statements is true about the visibility of local values in Terraform modules?
A. True
B. False
Answer: (SHOW ANSWER)
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: 88
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
Answer: B (LEAVE A REPLY)
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.

NEW QUESTION: 89
What type of Terraform component is used to fetch information from external providers, such as AMI IDs and network info, that can be utilized by other resources?
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: 90
You want to use API tokens and other secrets within your team's Terraform workspaces.
Where does HashiCorp recommend you store these sensitive values? (Pick the 3 correct responses)
A. In an HCP Terraform/Terraform Cloud variable, with the sensitive option checked.

B. In HashiCorp Vault.
C. In a terraform.tfvars file, securely managed and shared with your team.
D. In a terraform.tfvars file, checked into your version control system.
E. In a plaintext document on a shared drive.
Answer: (SHOW ANSWER)
Sensitive values such as API tokens should be stored in a secure way, either in Terraform Cloud variables marked as sensitive or in HashiCorp Vault. Storing secrets in version control systems or plaintext files is not recommended.
References:
Terraform Cloud Environment Variables

NEW QUESTION: 91
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: B (LEAVE A REPLY)
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: 92
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, but the 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: 93
One cloud block always maps to a single HCP Terraform/Terraform Cloud workspace.
A. True
B. False
Answer: A (LEAVE A REPLY)
Detailed Explanation:
* Rationale for Correct Answer (True):A cloud block in Terraform configuration specifies a single Terraform Cloud or HCP Terraform workspace. You cannot use one cloud block for multiple workspaces.
* Analysis of Incorrect Option:
* False: Incorrect because a cloud block is a one-to-one mapping with a single workspace.
* Key Concept:Cloud blocks manage remote operations and backend configuration tied to one workspace.
Reference:Terraform Exam Objective - Manage Terraform Workspaces and Cloud.

NEW QUESTION: 94
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
Answer: A (LEAVE A REPLY)
If you update the version constraint in your Terraform configuration, Terraform will update your lock file the next time you run terraform init3. This will ensure that you use the same provider versions across different machines and runs.

NEW QUESTION: 95
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: 96
Changing the Terraform backend from the default "local" backend to a different one after performing your first terrafom apply is:
A. Optional
B. Impossible
C. Mandatory
D. Discouraged
Answer: D (LEAVE A REPLY)
Changing the Terraform backend after performing the initial terraform apply is technically possible but strongly discouraged. This is because changing backends can lead to complexities in state management, requiring manual intervention such as state migration to ensure consistency. Terraform's documentation and best practices advise planning the backend configuration carefully before applying Terraform configurations to avoid such changes.References = This guidance is consistent with Terraform's official documentation, which recommends careful consideration and planning of backend configurations to avoid the need for changes.

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

NEW QUESTION: 98
You add a new provider to your configuration and immediately run terraform apply in the CD using the local backend. Why does the apply fail?
A. The Terraform CD needs you to log into Terraform Cloud first
B. Terraform requires you to manually run terraform plan first
C. Terraform needs to install the necessary plugins first
D. Terraform needs you to format your code according to best practices first
Answer: (SHOW ANSWER)
The reason why the apply fails after adding a new provider to the configuration and immediately running terraform apply in the CD using the local backend is because Terraform needs to install the necessary plugins first. Terraform providers are plugins that Terraform uses to interact with various cloud services and other APIs. Each provider has a source address that determines where to downloadit from. When Terraform encounters a new provider in the configuration, it needs to run terraform init first to install the provider plugins in a local directory. Without the plugins, Terraform cannot communicate with the provider and perform the desired actions.
References = [Provider Requirements], [Provider Installation]

NEW QUESTION: 99
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: 100

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: A,C,D (LEAVE A REPLY)

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: 101

A module can always refer to all variables declared in its parent module.

A. True

B. False

Answer: B (LEAVE A REPLY)

A module cannot always refer to all variables declared in its parent module, as it needs to explicitly declare input variables and assign values to them from the parent module's arguments. A module cannot access the parent module's variables directly, unless they are passed as input arguments.

NEW QUESTION: 102

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: 103
Which of the following arguments are required when declaring a Terraform output?
A. value
B. description
C. default
D. sensitive
Answer: (SHOW ANSWER)
When declaring a Terraform output, the value argument is required. Outputs are a way to extract information from Terraform-managed infrastructure, and the value argument specifies what data will be outputted. While other arguments like description and sensitive can provide additional context or security around the output, value is the only mandatory argument needed to define an output.References = The requirement of the value argument for outputs is specified in Terraform's official documentation, which provides guidelines on defining and using outputs in Terraform configurations.

NEW QUESTION: 104
What does the default "local" Terraform backend store?
A. tfplan files
B. State file
C. Provider plugins
D. Terraform binary
Answer: B (LEAVE A REPLY)
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: 105
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
Answer: D (LEAVE A REPLY)
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.

NEW QUESTION: 106
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: 107
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
Answer: D (LEAVE A REPLY)
The key principle of infrastructure as code that is not listed among the options is golden 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.
Versioned infrastructure: This means that the infrastructure code is stored in a version control system, such as Git, that tracks the changes and history of the code. This makes the infrastructure code reusable, auditable, and collaborative, and enables practices such as branching, merging, and rollback. References = [Introduction to Infrastructure as Code with Terraform], [Infrastructure as Code in a Private or Public Cloud]

NEW QUESTION: 108
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: (SHOW ANSWER)
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: 109
What is the best practice when making changes to infrastructure as code (IaC) in a team environment to ensure collaboration, traceability, and automation?
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: (SHOW ANSWER)
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: 110
It is best practice to store secret data in the same version control repository as your Terraform configuration.
A. True
B. False
Answer: (SHOW ANSWER)
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.

NEW QUESTION: 111
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: 112
Which Terraform function would you use to read the contents of a file, such as a public key file (id_rsa.pub), into 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: 113
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
Answer: A (LEAVE A REPLY)
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 run terraform init -upgrade to update them.

NEW QUESTION: 114
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: 115
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
Answer: F (LEAVE A REPLY)
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. References =
*1: Providers - Configuration Language | Terraform | HashiCorp Developer
*2: Plugin Development - How Terraform Works With Plugins | Terraform | HashiCorp Developer
*3: Terraform Registry
*: Terraform Registry

NEW QUESTION: 116
When you initialize Terraform, where does it cache modules from the public Terraform Registry?
A. In the /tmp directory.
B. In the .terraform sub-directory.
C. In memory.
D. They are not cached.
Answer: B (LEAVE A REPLY)
Terraformdownloads modules and providersinto the .terraform directory inside the working directory.
A (/tmp/)- Incorrect; modules arenot stored temporarily.
C (In memory)- Incorrect; modules persist on disk.
D (Not cached)- Incorrect; Terraformdoes cache modulesfor efficiency.
Official Terraform Documentation Reference:
Terraform Module Caching

NEW QUESTION: 117
Which of the following is not a valid Terraform variable type?
A. list
B. array
C. nap
D. string
Answer: (SHOW ANSWER)
This is not a valid Terraform variable type. The other options are valid variable types that can store different kinds of values2.

NEW QUESTION: 118
You have a Terraform configuration that defines a single virtual machine with no references to it, You have run terraform apply to create the resource, and then removed the resource definition from your Terraform configuration file.
What will happen you run terraform apply in the working directory again?
A. Terraform will remove the virtual machine from the state file, but the resource will still exist
B. Nothing
C. Terraform will error
D. Terraform will destroy the virtual machine
Answer: D (LEAVE A REPLY)
This is what will happen if you run terraform apply in the working directory again, after removing the resource definition from your Terraform configuration file. Terraform will detect that there is a resource in the state file that is not present in the configuration file, and will assume that you want to delete it.

NEW QUESTION: 119
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
Answer: B (LEAVE A REPLY)
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. References = [Terraform State]

NEW QUESTION: 120
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: 121
Which of these actions will prevent two Terraform runs from changing the same state file at the same time?
A. Refresh the state after running Terraform
B. Delete the state before running Terraform
C. Configure state locking for your state backend
D. Run Terraform with parallelism set to 1
Answer: (SHOW ANSWER)
To prevent two Terraform runs from changing the same state file simultaneously, state locking is used. State locking ensures that when one Terraform operation is running, others will be blocked from making changes to the same state, thus preventing conflicts and data corruption. This is achieved by configuring the state backend to support locking, which will lock the state for all operations that could write to the state.References
= This information is supported by Terraform's official documentation, which explains the importance of state locking and how it can be configured for different backends to prevent concurrent state modifications .

NEW QUESTION: 122
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: 123
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.