HashiCorp.TA-002-P.v2022-09-14.q200

Exam Code:	TA-002-P
Exam Name:	HashiCorp Certified: Terraform Associate
Certification Provider:	HashiCorp
Free Question Number:	200
Version:	v2022-09-14
# of views:	541
# of Questions views:	5747
https://www.dumpsfiles.com/files/HashiCorp/TA-002-P/HashiCorp.TA-002-P.v2022-09-14.q200	

NEW QUESTION: 1
You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.
What are the two things you must do to achieve this? (Choose two.)
A. Provision new VMs using Terraform with the same VM names
B. Use the terraform import command for the existing VMs
C. Write Terraform configuration for the existing VMs
D. Run the terraform import-gcp command
Answer: (SHOW ANSWER)
The terraform import command is used to import existing infrastructure. Import existing Google Cloud resources into Terraform with Terraformer.

NEW QUESTION: 2
What is the result of the following terraform function call?
A. hello
B. what?
C. goodbye
Answer: B (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/configuration/functions/lookup.html

NEW QUESTION: 3
Which option can not be used to keep secrets out of Terraform configuration files?
A. Environment variables
B. A -var flag
C. A Terraform provider
D. secure string

Answer: C (LEAVE A REPLY)

NEW QUESTION: 4
What is one disadvantage of using dynamic blocks in Terraform?
A. They cannot be used to loop through a list of values
B. Dynamic blocks can construct repeatable nested blocks
C. Terraform will run more slowly
D. They make configuration harder to read and understand
Answer: A (LEAVE A REPLY)

NEW QUESTION: 5
Given the Terraform configuration below, in which order will the resources be created?
1. resource "aws_instance" "web_server"
2. {
3. ami = "ami-b374d5a5"
4. instance_type = "t2.micro"
5. }
6. resource "aws_eip" "web_server_ip"
7. {
8. vpc = true instance = aws_instance.web_server.id
9. }
A. aws_eip will be created first
aws_instance will be created second
B. aws_eip will be created first
aws_instance will be created second
C. Resources will be created simultaneously
D. aws_instance will be created first
aws_eip will be created second
Answer: (SHOW ANSWER)
Implicit and Explicit Dependencies
By studying the resource attributes used in interpolation expressions, Terraform can automatically infer when one resource depends on another. In the example above, the reference to aws_instance.web_server.id creates an implicit dependency on the aws_instance named web_server.
Terraform uses this dependency information to determine the correct order in which to create the different resources.
# Example of Implicit Dependency
resource "aws_instance" "web_server" {
ami = "ami-b374d5a5"
instance_type = "t2.micro"
}
resource "aws_eip" "web_server_ip" {
vpc = true

instance = aws_instance.web_server.id
}
In the example above, Terraform knows that the aws_instance must be created before the aws_eip.
Implicit dependencies via interpolation expressions are the primary way to inform Terraform about these relationships, and should be used whenever possible.
Sometimes there are dependencies between resources that are not visible to Terraform. The depends_on argument is accepted by any resource and accepts a list of resources to create explicit dependencies for.
For example, perhaps an application we will run on our EC2 instance expects to use a specific Amazon S3 bucket, but that dependency is configured inside the application code and thus not visible to Terraform. In that case, we can use depends_on to explicitly declare the dependency:
# Example of Explicit Dependency
# New resource for the S3 bucket our application will use.
resource "aws_s3_bucket" "example" {
bucket = "terraform-getting-started-guide"
acl = "private"
}
# Change the aws_instance we declared earlier to now include "depends_on" resource "aws_instance" "example" { ami = "ami-2757f631" instance_type = "t2.micro"
# Tells Terraform that this EC2 instance must be created only after the
# S3 bucket has been created.
depends_on = [aws_s3_bucket.example]
}
https://learn.hashicorp.com/terraform/getting-started/dependencies.html

NEW QUESTION: 6
When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)
A. directly querying HashiCorp Vault for the secrets
B. command-line key/value pairs
C. use the -backend-config=PATH to specify a separate config file
D. interactively on the command line
Answer: B,C,D (LEAVE A REPLY)
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.
With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments: https://www.terraform.io/docs/backends/init.html#backend-initialization

NEW QUESTION: 7
What is not processed when running a terraform refresh?

A. State file
B. Configuration file
C. Credentials
D. Cloud provider
Answer: C,D (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/cli/commands/refresh.html

NEW QUESTION: 8
Which of the following is the right substitute for static values that can make Terraform configuration file more dynamic and reusable?
A. Output value
B. Input parameters
C. Functions
D. Modules
Answer: (SHOW ANSWER)
Explanation
Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.

NEW QUESTION: 9
What happens when a terraform apply command is executed?
A. Creates the execution plan for the deployment of resources.
B. Applies the changes required in the target infrastructure in order to reach the desired configuration.
C. The backend is initialized and the working directory is prepped.
D. Reconciles the state Terraform knows about with the real-world infrastructure.
Answer: B (LEAVE A REPLY)
The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.
https://www.terraform.io/docs/commands/apply.html

NEW QUESTION: 10
In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?
A. Resource dependencies are identified and maintained in a file called resource.dependencies. Each terraform provider is required to maintain a list of all resource dependencies for the provider and it's included with the plugin during initialization when terraform init is executed. The file is located in the terraform.d folder.
B. The terraform binary contains a built-in reference map of all defined Terraform resource dependencies. Updates to this dependency map are reflected in terraform versions. To ensure you are working with the latest resource dependency map you much be running the latest version of Terraform.
C. Resource dependencies are handled automatically by the depends_on meta_argument, which is set to true by default.
D. Terraform analyses any expressions within a resource block to find references to other objects, and treats those references as implicit ordering requirements when creating, updating, or destroying resources.

Answer: (SHOW ANSWER)
https://www.terraform.io/docs/configuration/resources.html

NEW QUESTION: 11
Select all Operating Systems that Terraform is available for. (select five)
A. Linux
B. macOS
C. Unix
D. Solaris
E. Windows
F. FreeBSD
Answer: A,B,D,E,F (LEAVE A REPLY)
Terraform is available for macOS, FreeBSD, OpenBSD, Linux, Solaris, Windows https://www.terraform.io/downloads.html

NEW QUESTION: 12
What is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault?
A. API access using the AppRole auth method
B. integration with a tool like Jenkins
C. CLI access from the same machine running Terraform
D. Vault provider
Answer: (SHOW ANSWER)

NEW QUESTION: 13
Why should secrets not be hard coded into Terraform code? Choose two correct answers
A. The Terraform code is copied to the target resources to be applied locally and could expose secrets if a target resource is compromised.
B. All passwords should be rotated on a quarterly basis.
C. It makes the code less reusable.
D. Terraform code is typically stored in version control, as well as copied to the systems from h it's run. Any of those may not have robust security mechanisms.
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 14
One remote backend configuration always maps to a single remote workspace.
A. False
B. True
Answer: B (LEAVE A REPLY)

NEW QUESTION: 15
Terraform works well in Windows but a Windows server is required.
A. False

B. True
Answer: (SHOW ANSWER)
You may see this
Terraform does not require GO language to be installed as a prerequisite and it does not require a Windows Server as well.

NEW QUESTION: 16
Once a resource is marked as tainted, the next plan will show that the resource will be _____ and _____ and the next apply will implement this change.
A. destroyed and not recreated
B. destroyed and recreated
C. recreated and tainted
D. tainted and not destroyed
Answer: B (LEAVE A REPLY)

NEW QUESTION: 17
ABC Enterprise has recently tied up with multiple small organizations for exchanging database information. Due to this, the firewall rules are increasing and are more than 100 rules. This is leading firewall configuration file that is difficult to manage. What is the way this type of configuration can be managed easily?
A. Terraform Backends
B. Terraform Functions
C. Dynamic Blocks
D. Terraform Expression
Answer: C (LEAVE A REPLY)

NEW QUESTION: 18
Which of the following is not a key principle of infrastructure as code?
A. Versioned infrastructure
B. Golden images
C. Idempotence
D. Self-describing infrastructure
Answer: A,B,D (LEAVE A REPLY)
Reference: https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as- code#:~:text=Idempotence%20is%20a%20principle%20of,of%20the%20environment's%20starting%20state.

NEW QUESTION: 19
Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)
A. Ubuntu
Explanation
https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements
B. CentOS
C. Amazon Linux
D. Red Hat
E. Unix
Answer: A,B,C,D (LEAVE A REPLY)

NEW QUESTION: 20
You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (*.tf files). You need to enable debug messages to find this out.
Which of the following would achieve this?
A. Set the environment variable TF_LOG_PATH
B. Set verbose logging for each provider in your Terraform configuration
C. Set the environment variable TF_LOG=TRACE
D. Set the environment variable TF_VAR_log=TRACE
Answer: C (LEAVE A REPLY)

NEW QUESTION: 21
Which of the following value will be accepted for my_var?
1. variable "my_var"
2. {
3. type = string
4. }
A. 15
B. "15"
C. Both A and B
D. None of the above
Answer: C (LEAVE A REPLY)
The Terraform language will automatically convert number and bool values to string values when needed, and vice-versa as long as the string contains a valid representation of a number or boolean value.
Example
* true converts to "true", and vice-versa
* false converts to "false", and vice-versa
* 15 converts to "15", and vice-versa
Where possible, Terraform automatically converts values from one type to another in order to produce the expected type. If this isn't possible, Terraform will produce a type mismatch error and you must update the configuration with a more suitable expression.

https://www.terraform.io/docs/configuration/expressions.html#type-conversion

NEW QUESTION: 22
Terraform init can indeed be run only a few times, because, every time terraform init will initialize the project
, and download all plugins from the internet repository , regardless of whether they were present or not , and this increases the waiting time
A. True
B. False
Answer: (SHOW ANSWER)
Explanation
Re-running init with modules already installed will install the sources for any modules that were added to configuration since the last init, but will not change any already-installed modules. Use -upgrade to override this behavior, updating all modules to the latest available source code.
https://www.terraform.io/docs/commands/init.html

NEW QUESTION: 23
Terraform can import modules from a number of sources - which of the following is not a valid source?
A. Local path
B. Terraform Module Registry
C. FTP server
D. GitHub repository
Answer: (SHOW ANSWER)

NEW QUESTION: 24
What feature of Terraform Cloud and/or Terraform Enterprise can you publish and maintain a set of custom modules which can be used within your organization?
A. remote runs
B. custom VCS integration
C. private module registry
D. Terraform registry
Answer: C (LEAVE A REPLY)

NEW QUESTION: 25
What value does the Terraform Cloud/Terraform Enterprise private module registry provide over the public Terraform Module Registry?
A. The ability to share modules with public Terraform users and members of Terraform Enterprise Organizations
B. The ability to tag modules by version or release
C. The ability to restrict modules to members of Terraform Cloud or Enterprise organizations
D. The ability to share modules publicly with any user of Terraform
Answer: (SHOW ANSWER)

Terraform Registry is an index of modules shared publicly using this protocol. This public registry is the easiest way to get started with Terraform and find modules created by others in the community.

NEW QUESTION: 26
Multiple providers can be declared within a single Terraform configuration file.
A. True
B. False
Answer: (SHOW ANSWER)
Explanation
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration.
For Example
# The default provider configuration
provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region
provider "aws" {
alias = "west"
region = "us-west-2"
}
The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION: 27
You have created a terraform script that uses a lot of new constructs that have been introduced in terraform v0.12. However, many developers who are cloning the script from your git repo, are using v0.11, and getting errors. What can be done from your end to solve this problem?
A. Force developer to use v0.12 by using terraform setting 'required_version' and set it to >=0.12.
B. Refactor the code to support both v0.11, and v0.12. It might be a difficult process, but there is no other way.
C. Add a condition in front of each such specific construct, to check whether the running terraform version id v0.11 or v0.12, and ,work accordingly.
D. Add comments in your code to tell developers to use v0.12 . If they use v0.11 , that should be their problem , which they need to figure out.
Answer: A (LEAVE A REPLY)
https://www.terraform.io/docs/configuration/terraform.html

NEW QUESTION: 28
You have 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 will be deleted by Terraform.
Which command should you use to show all of the resources that will be deleted? (Choose two.)
A. Run terraform plan -destroy.
B. This is not possible. You can only show resources that will be created.
C. Run terraform state rm *.
D. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval.
Answer: C,D (LEAVE A REPLY)
Explanation/Reference: https://www.terraform.io/docs/cli/commands/state/rm.html

NEW QUESTION: 29
Which of the below features of Terraform can be used for managing small differences between different environments which can act more like completely separate working directories.
A. Repositories
B. Workspaces
C. Environment Variables
D. Backends
Answer: (SHOW ANSWER)
workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. They are convenient in a number of situations, but cannot solve all problems.
A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. For example, a developer working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.
Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure. When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.
https://www.terraform.io/docs/state/workspaces.html
https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces

NEW QUESTION: 30
Which of the following is not true of Terraform providers?
A. Some providers are maintained by HashiCorp
B. None of the above
C. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers
D. Providers can be maintained by a community of users
E. Providers can be written by individuals

Answer: C (LEAVE A REPLY)

NEW QUESTION: 31
Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.
A. terraform apply
B. terraform output
C. terraform validate
D. terraform state
E. terraform plan
Answer: A,E (LEAVE A REPLY)

NEW QUESTION: 32
Hanah is writing a terraform configuration with nested modules, there are multiple places where she has to use the same conditional expression but she wants to avoid repeating the same values or expressions multiple times in the configuration,. What is a better approach to dealing with this?
A. Expressions
B. Local Values
C. Variables
D. Functions
Answer: B (LEAVE A REPLY)
https://www.terraform.io/docs/configuration/locals.html

NEW QUESTION: 33
You want to use terraform import to start managing infrastructure that was not originally provisioned through infrastructure as code. Before you can import the resource's current state, what must you do in order to prepare to manage these resources using Terraform?
A. Run terraform refresh to ensure that the state file has the latest information for existing resources.
B. Update the configuration file to include the new resources.
C. Shut down or stop using the resources being imported so no changes are inadvertently missed.
D. Modify the Terraform state file to add the new resources.
Answer: B (LEAVE A REPLY)

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

NEW QUESTION: 34
Terraform and Terraform providers must use the same major version number in a single configuration.
A. True
B. False
Answer: B (LEAVE A REPLY)

NEW QUESTION: 35
When writing Terraform code, HashiCorp recommends that you use how many spaces between each nesting level?
A. 0
B. 1
C. 2
D. 4
Answer: (SHOW ANSWER)
The Terraform parser allows you some flexibility in how you lay out the elements in your configuration files, but the Terraform language also has some idiomatic style conventions which we recommend users always follow for consistency between files and modules written by different teams. Automatic source code formatting tools may apply these conventions automatically.
Indent two spaces for each nesting level.
When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs:
ami = "abc123"

instance_type = "t2.micro"
When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them. Use one blank line to separate the arguments from the blocks.
Use empty lines to separate logical groups of arguments within a block.
For blocks that contain both arguments and "meta-arguments" (as defined by the Terraform language semantics), list meta-arguments first and separate them from other arguments with one blank line. Place meta-argument blocks last and separate them from other blocks with one blank line.
resource "aws_instance" "example" {
count = 2 # meta-argument first
ami = "abc123"
instance_type = "t2.micro"
network_interface {
# ...
}
lifecycle { # meta-argument block last
create_before_destroy = true
}
}
Top-level blocks should always be separated from one another by one blank line. Nested blocks should also be separated by blank lines, except when grouping together related blocks of the same type (like multiple provisioner blocks in a resource).
Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example: root_block_device, ebs_block_device and ephemeral_block_device on aws_instance form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

NEW QUESTION: 36
does not require GO language to be installed as a prerequisite and it does not require a Windows Server as well.
A. False
B. True
Answer: A (LEAVE A REPLY)

NEW QUESTION: 37
What command should you run to display all workspaces for the current configuration?
A. terraform workspace
B. terraform workspace show
C. terraform workspace list
D. terraform show workspace
Answer: C (LEAVE A REPLY)
Explanation
terraform workspace list
The command will list all existing workspaces.
Reference: https://www.terraform.io/docs/cli/commands/workspace/list.html

NEW QUESTION: 38
You write a new Terraform configuration and immediately run terraform apply in the CLI using the local backend.
Why will the apply fail?
A. Terraform needs you to format your code according to best practices first
B. Terraform requires you to manually run terraform plan first
C. The Terraform CLI needs you to log into Terraform cloud first
D. Terraform needs to install the necessary plugins first
Answer: C (LEAVE A REPLY)

NEW QUESTION: 39
Where does the Terraform local backend store its state?
A. In the /tmp directory
B. In the terraform.tfvars file
C. In the terraform.tfstate file
D. In the user's .terraformrc file
Answer: C (LEAVE A REPLY)
The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.

NEW QUESTION: 40
Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)
A. Destroy
B. Apply
C. Import
D. Init
E. Validate
Answer: B,D (LEAVE A REPLY)
Reference: https://www.terraform.io/guides/core-workflow.html

NEW QUESTION: 41
State is a requirement for Terraform to function
A. True
B. False
Answer: (SHOW ANSWER)
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run.
Purpose of Terraform State
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.

As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).

1. Mapping to the Real World

Terraform requires some sort of database to map Terraform config to the real world. When you have a resource resource "aws_instance" "foo" in your configuration, Terraform uses this map to know that instance i- abcd1234 is represented by that resource.

For some providers like AWS, Terraform could theoretically use something like AWS tags. Early prototypes of Terraform actually had no state files and used this method. However, we quickly ran into problems. The first major issue was a simple one: not all resources support tags, and not all cloud providers support tags.

Therefore, for mapping configuration to resources in the real world, Terraform uses its own state structure.

2. Metadata

Alongside the mappings between resources and remote objects, Terraform must also track metadata such as resource dependencies.

Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.

To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.

One way to avoid this would be for Terraform to know a required ordering between resource types. For example, Terraform could know that servers must be deleted before the subnets they are a part of. The complexity for this approach quickly explodes, however: in addition to Terraform having to understand the ordering semantics of every resource for every cloud, Terraform must also understand the ordering across providers.

Terraform also stores other metadata for similar reasons, such as a pointer to the provider configuration that was most recently used with the resource in situations where multiple aliased providers are present.

3. Performance

In addition to basic mapping, Terraform stores a cache of the attribute values for all resources in the state. This is the most optional feature of Terraform state and is done only as a performance improvement.

When running a terraform plan, Terraform must know the current state of resources in order to effectively determine the changes that it needs to make to reach your desired configuration.

For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.

4. Syncing

In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but when using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects.

Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION: 42
HashiCorp offers multiple versions of Terraform, including Terraform open-source, Terraform Cloud, and Terraform Enterprise. Which of the following Terraform features are only available in the Enterprise edition? (select four)
A. SAML/SSO
B. Sentinel
C. Audit Logs
D. Clustering
E. Private Module Registry
F. Private Network Connectivity
Answer: (SHOW ANSWER)
Explanation
While there are a ton of features that are available to open source users, many features that are part of the Enterprise offering are geared towards larger teams and enterprise functionality. To see what specific features are part of Terraform Cloud and Terraform Enterprise, check out this link. https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION: 43
You want to use different AMI images for different regions and for the purpose you have defined following code block.
1. variable "images"
2. {
3. type = "map"
4.
5. default = {
6. us-east-1 = "image-1234"
7. us-west-2 = "image-4567"
8. us-west-1 = "image-4589"
9. }
10. }
What of the following approaches needs to be followed in order to select image-4589?
A. lookup(var.images["us-west-1"]
B. var.images[3]
C. var.images[2]
D. var.images["us-west-1"]
Answer: (SHOW ANSWER)

NEW QUESTION: 44
In the following code snippet, the block type is identified by which string?
A. "db"

B. "aws_instance"
C. instance_type
D. resource
Answer: D (LEAVE A REPLY)

NEW QUESTION: 45
A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?
A. terraform workspace dev
B. terraform workspace select dev
C. terraform workspace -switch dev
D. terraform workspace switch dev
Answer: (SHOW ANSWER)
Explanation
The terraform workspace select command is used to choose a different workspace to use for further operations.
https://www.terraform.io/docs/commands/workspace/select.html

NEW QUESTION: 46
Which flag would be used within a Terraform configuration block to identify the specific version of a provider required?
A. required-provider
B. required-version
C. required_providers
D. required_versions
Answer: C (LEAVE A REPLY)
For production use, you should constrain the acceptable provider versions via configuration file to ensure that new versions with breaking changes will not be automatically installed by terraform init in the future.
Example
terraform {
required_providers {
aws = ">= 2.7.0"
}
}

NEW QUESTION: 47
What are some of the problems of how infrastructure was traditionally managed before Infrastructure as Code?
(select three)
A. Requests for infrastructure or hardware required a ticket, increasing the time required to deploy applications
B. Traditional deployment methods are not able to meet the demands of the modern business where resources tend to live days to weeks, rather than months to years
C. Traditionally managed infrastructure can't keep up with cyclic or elastic applications
D. Pointing and clicking in a management console is a scalable approach and reduces human error as businesses are moving to a multi-cloud deployment model
Answer: A,B,C (LEAVE A REPLY)
Explanation
Businesses are making a transition where traditionally-managed infrastructure can no longer meet the demands of today's businesses. IT organizations are quickly adopting the public cloud, which is predominantly API-driven. To meet customer demands and save costs, application teams are architecting their applications to support a much higher level of elasticity, supporting technology like containers and public cloud resources.
These resources may only live for a matter of hours; therefore the traditional method of raising a ticket to request resources is no longer a viable option Pointing and clicking in a management console is NOT scale and increases the change of human error.

NEW QUESTION: 48
When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)
1. terraform
2. {
3. required_providers
4. {
5. aws = "~> 1.2.0"
6. }
7. }
A. 1.3.1
B. 1.2.3
C. 1.2.9
D. 1.3.0
Answer: (SHOW ANSWER)
Explanation
As your Terraform usage becomes more advanced, there are some cases where you may need to modify the Terraform state. Rather than modify the state directly, the terraform state commands can be used in many cases instead. This command is a nested subcommand, meaning that it has further subcommands.
https://www.terraform.io/docs/commands/state/index.html

NEW QUESTION: 49
Terraform Enterprise (also referred to as pTFE) requires what type of backend database for a clustered deployment?

A. PostgreSQL
B. Cassandra
C. MySQL
D. MSSQL
Answer: A (LEAVE A REPLY)
Explanation
External Services mode stores the majority of the stateful data used by the instance in an external PostgreSQL database and an external S3-compatible endpoint or Azure blob storage. There is still critical data stored on the instance that must be managed with snapshots. Be sure to check the PostgreSQL Requirements for information that needs to be present for Terraform Enterprise to work. This option is best for users with expertise managing PostgreSQL or users that have access to managed PostgreSQL offerings like AWS RDS.

NEW QUESTION: 50
Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.
A. False
B. True
Answer: B (LEAVE A REPLY)
Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.
https://www.terraform.io/docs/state/sensitive-data.html#recommendations

NEW QUESTION: 51
Which task does teraform ini- not perform?
A. Sources all providers present in the configuration and ensures they are downloaded and available locally
B. Connects to the backend
C. Sources any modules and copies the configuration locally
D. Validates all required variables are present
Answer: D (LEAVE A REPLY)

NEW QUESTION: 52
Which of the following is the correct way to pass the value in the variable num_servers into a module with the input servers?
A. servers = num_servers
B. servers = var.num_servers
C. servers = var(num_servers)
D. servers = variable.num_servers
Answer: (SHOW ANSWER)

NEW QUESTION: 53

You have configured an Auto Scaling group in AWS to automatically scale the number of instances behind a load balancer based on the instances CPU utilization. The instances are configured using a Launch Configuration. You have observed that the Auto Scaling group doesn't successfully scale when you apply changes that require replacing the Launch Configuration. Why is this happening?

A. You need to configure an explicit dependency for the Auto Scaling group using the depends_on meta-parameter.

B. You need to configure an explicit dependency for the Launch Configuration using the depends_on meta-parameter.

C. You need to configure the Auto Scaling group's create_before_destroy meta-parameter.

D. You need to configure the Launch Configuration's create_before_destroy meta-parameter.

Answer: D (LEAVE A REPLY)

https://www.terraform.io/docs/providers/aws/r/launch_configuration.html#using-withautoscaling-groups

NEW QUESTION: 54

What is the name assigned by Terraform to reference this resource?

```
resource "azurerm_resource_group" "dev" {
  name = "test"
  location = "westus"
}
```

A. azurerm_resource_group

B. dev

C. azurerm

D. test

Answer: (SHOW ANSWER)

NEW QUESTION: 55

Multiple providers can be declared within a single Terraform configuration file.

A. True

B. False

Answer: (SHOW ANSWER)

You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration.

For Example

```
# The default provider configuration
provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region
provider "aws" {
alias = "west"
region = "us-west-2"
```

}
The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION: 56
When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?
A. Object Storage
B. State Locking
C. WorkSpaces
D. Encryption
Answer: B (LEAVE A REPLY)
If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state.
State locking happens automatically on all operations that could write state. You won't see any message that it is happening. If state locking fails, Terraform will not continue. You can disable state locking for most commands with the -lock flag but it is not recommended.
If acquiring the lock is taking longer than expected, Terraform will output a status message. If Terraform doesn't output a message, state locking is still occurring if your backend supports it.
Not all backends support locking. Please view the list of backend types for details on whether a backend supports locking or not.
https://www.terraform.io/docs/state/locking.html

NEW QUESTION: 57
A terraform apply can not _____ infrastructure.
A. destroy
B. import
C. provision
D. change
Answer: D (LEAVE A REPLY)

NEW QUESTION: 58
In Terraform 0.13 and above, outside of the required_providers block, Terraform configurations always refer to providers by their local names.
A. True
B. False
Answer: A (LEAVE A REPLY)
Outside of the required_providers block, Terraform configurations always refer to providers by their local names.

NEW QUESTION: 59

What value does the Terraform Cloud/Terraform Enterprise private module registry provide over the public Terraform Module Registry?

A. The ability to share modules with public Terraform users and members of Terraform Enterprise Organizations

B. The ability to tag modules by version or release

C. The ability to restrict modules to members of Terraform Cloud or Enterprise organizations

D. The ability to share modules publicly with any user of Terraform

Answer: D (LEAVE A REPLY)

Explanation

Terraform Registry is an index of modules shared publicly using this protocol. This public registry is the easiest way to get started with Terraform and find modules created by others in the community.

Reference: https://www.terraform.io/docs/language/modules/sources.html

NEW QUESTION: 60

Module variable assignments are inherited from the parent module and do not need to be explicitly set.

A. True

B. False

Answer: B (LEAVE A REPLY)

NEW QUESTION: 61

A provider configuration block is required in every Terraform configuration.

Example:

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

A. True

B. False

Answer: A (LEAVE A REPLY)

Reference: https://github.com/hashicorp/terraform/issues/17928

NEW QUESTION: 62

The Security Operations team of ABC Enterprise wants to mandate that all the Terraform configuration that creates an S3 bucket must have encryption feature enabled. What is the best way to achieve it?

A. Use Sentinel Policies.

B. Use S3 bucket policy.
C. Create a script that checks the encryption parameter is enabled on every git commit.
D. Shared a SOP to engineers to mandate encryption feature on S3.
Answer: A (LEAVE A REPLY)
Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.
Using Sentinel with Terraform Cloud involves:
* Defining the policies - Policies are defined using the policy language with imports for parsing the Terraform plan, state and configuration.
* Managing policies for organizations - Users with permission to manage policies can add policies to their organization by configuring VCS integration or uploading policy sets through the API. They also define which workspaces the policy sets are checked against during runs. (More about permissions.)
* Enforcing policy checks on runs - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is executed.
* Mocking Sentinel Terraform data - Terraform Cloud provides the ability to generate mock data for any run within a workspace. This data can be used with the Sentinel CLI to test policies before deployment.
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION: 63
Only the user that generated a plan may apply it.
A. True
B. False
Answer: A (LEAVE A REPLY)
Explanation
The optional -out argument can be used to save the generated plan to a file for later execution with terraform apply, which can be useful when running Terraform in automation.
Reference: https://learn.hashicorp.com/tutorials/terraform/automate-terraform

NEW QUESTION: 64
A "backend" in Terraform determines how state is loaded and how an operation such as apply is executed. Which of the following is not a supported backend type?
A. Terraform enterprise
B. Consul
C. Github
D. S3
E. Artifactory
Answer: C (LEAVE A REPLY)
Github is not a supported backend type.
https://www.terraform.io/docs/backends/types/index.html

NEW QUESTION: 65

A single terraform resource file that defines an aws_instance resource can simply be renamed to vsphere_virtual_machine in order to switch cloud providers.

A. True

B. False

Answer: (SHOW ANSWER)

Every provider has its own required and allowed declarations none of which match between cloud providers.

NEW QUESTION: 66

When using multiple configurations of the same Terraform provider, what meta-argument must be included in any non-default provider configurations?

A. name

B. depends_on

C. alias

D. id

Answer: C (LEAVE A REPLY)

NEW QUESTION: 67

What is the result of the following terraform function call?

A. True

B. False

Explanation

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

Answer: B (LEAVE A REPLY)

NEW QUESTION: 68

The terraform state command can be used to ____

A. Update current state

B. Refresh existing state file

C. Print the current state file in console

D. It is not a valid command

Answer: A (LEAVE A REPLY)

The terraform state command is used for advanced state management. Rather than modify the state directly, the terraform state commands can be used in many cases instead.

https://www.terraform.io/docs/commands/state/index.html

NEW QUESTION: 69

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

A. source = "git::https://example.com/my_test_module.git@tag=v1.0.4"

B. source = "git::https://example.com/my_test_module.git&ref=v1.0.4"

C. source = "git::https://example.com/my_test_module.git#tag=v1.0.4"
D. source = "git::https://example.com/my_test_module.git?ref=v1.0.4"
Answer: D (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/modules/sources.html#selecting-a-revision

NEW QUESTION: 70
ABC Enterprise has recently tied up with multiple small organizations for exchanging database information.
Due to this, the firewall rules are increasing and are more than 100 rules. This is leading firewall configuration file that is difficult to manage. What is the way this type of configuration can be managed easily?
A. Terraform Expression
B. Dynamic Blocks
C. Terraform Backends
D. Terraform Functions
Answer: B (LEAVE A REPLY)

NEW QUESTION: 71
The terraform init command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing configuration or state.
A. False
B. True
Answer: B (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/commands/init.html

NEW QUESTION: 72
In Terraform Enterprise, a workspace can be mapped to how many VCS repos?
A. 5
B. 2
C. 3
D. 1
Answer: (SHOW ANSWER)
Explanation
A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.
https://www.terraform.io/docs/cloud/workspaces/vcs.html

NEW QUESTION: 73
Which of the following commands will launch the Interactive console for Terraform interpolations?
A. terraform console
B. terraform cli
C. terraform cmdline

Explanation
https://www.terraform.io/docs/commands/console.html
D. terraform
Answer: B (LEAVE A REPLY)

NEW QUESTION: 74
After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?
A. ERROR
B. INFO
C. WARN
D. TRACE
E. DEBUG
Answer: D (LEAVE A REPLY)
Explanation
Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value.
This will cause detailed logs to appear on stderr.
You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.
TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name.
Examples:
export TF_LOG=DEBUG
export TF_LOG=TRACE

NEW QUESTION: 75
Terraform variables and outputs that set the "description" argument will store that description in the state file.
A. False
B. True
Answer: A (LEAVE A REPLY)

NEW QUESTION: 76
Which provisioner invokes a process on the resource created by Terraform?
A. remote-exec
B. null-exec
C. local-exec
D. file
Answer: A (LEAVE A REPLY)
Explanation
The remote-exec provisioner invokes a script on a remote resource after it is created. Reference:
https://www.terraform.io/docs/language/resources/provisioners/remote-exec.html

NEW QUESTION: 77
Terraform has detailed logs which can be enabled by setting the _____ environmental variable.
A. TF_TRACE
B. TF_DEBUG
C. TF_LOG
D. TF_INFO
Answer: C (LEAVE A REPLY)
Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr.
You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.
TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name.
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 78
Which statements best describes what the local variable assignment is doing in the following code snippet:
A. Create a list of route table names eliminating duplicates
B. Create a map of route table names from a list of subnet names
C. Create a distinct list of route table name objects
D. Create a map of route table names to subnet names
Answer: (SHOW ANSWER)

NEW QUESTION: 79
Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?
A. Use terraform console command to have an interactive UI with full access to the underlying terraform state to run your interpolations , and debug at real-time.
B. Add a breakpoint in your code, using the watch keyword , and output the value to console for temporary debugging.
C. Use terraform zipmap function , it will be able to easily do the interpolations without complex code.
D. Use terraform console command to have an interactive UI , but you can only use it with local state , and it does not work with remote state.
Answer: A (LEAVE A REPLY)
Explanation

The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.
https://www.terraform.io/docs/commands/console.html

NEW QUESTION: 80
In order to reduce the time it takes to provision resources, Terraform uses parallelism. By default, how many resources will Terraform provision concurrently?
A. 10
B. 50
C. 20
D. 5
Answer: A (LEAVE A REPLY)

NEW QUESTION: 81
If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?
A. Manually update the state fire
B. Run terraform import
C. It will happen automatically
D. Run terraform refresh
Answer: C (LEAVE A REPLY)

NEW QUESTION: 82
A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?
A. terraform workspace dev
B. terraform workspace select dev
C. terraform workspace -switch dev
D. terraform workspace switch dev
Answer: (SHOW ANSWER)
Explanation
The terraform workspace select command is used to choose a different workspace to use for further operations.
https://www.terraform.io/docs/commands/workspace/select.html

NEW QUESTION: 83
If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?
A. Run terraform import
B. Run terraform refresh
C. Manually update the state fire
D. It will happen automatically
Answer: B (LEAVE A REPLY)

NEW QUESTION: 84
Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?
A. Use terraform console command to have an interactive UI with full access to the underlying terraform state to run your interpolations , and debug at real-time.
B. Add a breakpoint in your code, using the watch keyword , and output the value to console for temporary debugging.
C. Use terraform zipmap function , it will be able to easily do the interpolations without complex code.
D. Use terraform console command to have an interactive UI , but you can only use it with local state , and it does not work with remote state.
Answer: A (LEAVE A REPLY)
The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.
https://www.terraform.io/docs/commands/console.html

NEW QUESTION: 85
How is the Terraform remote backend different than other state backends such as S3, Consul, etc.?
A. It can execute Terraform runs on dedicated infrastructure on premises or in Terraform Cloud
B. It doesn't show the output of a terraform apply locally
C. It is only available to paying customers
D. All of the above
Answer: A (LEAVE A REPLY)
Explanation
If you and your team are using Terraform to manage meaningful infrastructure, we recommend using the remote backend with Terraform Cloud or Terraform Enterprise.
Reference: https://www.terraform.io/docs/language/settings/backends/index.html

NEW QUESTION: 86
You cannot publish your own modules on the Terraform Registry.
A. False
B. True
Answer: A (LEAVE A REPLY)
https://www.terraform.io/docs/registry/modules/publish.html

NEW QUESTION: 87
After executing a terraform apply, you notice that a resource has a tilde (~) next to it. What does this infer?
A. The resource will be updated in place.
B. The resource will be created.
C. Terraform can't determine how to proceed due to a problem with the state file.
D. The resource will be destroyed and recreated.
Answer: A (LEAVE A REPLY)

Explanation
The prefix -/+ means that Terraform will destroy and recreate the resource, rather than updating it in-place.
The prefix ~ means that some attributes and resources can be updated in-place.
$ terraform apply
aws_instance.example: Refreshing state... [id=i-0bbf06244e44211d1]
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement
Terraform will perform the following actions:
# aws_instance.example must be replaced
-/+ resource "aws_instance" "example" {
~ ami = "ami-2757f631" -> "ami-b374d5a5" # forces replacement
~ arn = "arn:aws:ec2:us-east-1:130490850807:instance/i-0bbf06244e44211d1" -> (known after apply)
~ associate_public_ip_address = true -> (known after apply)
~ availability_zone = "us-east-1c" -> (known after apply)
~ cpu_core_count = 1 -> (known after apply)
~ cpu_threads_per_core = 1 -> (known after apply)
- disable_api_termination = false -> null
- ebs_optimized = false -> null
get_password_data = false
+ host_id = (known after apply)
~ id = "i-0bbf06244e44211d1" -> (known after apply)
~ instance_state = "running" -> (known after apply)
instance_type = "t2.micro"
~ ipv6_address_count = 0 -> (known after apply)
~ ipv6_addresses = [] -> (known after apply)
+ key_name = (known after apply)
- monitoring = false -> null
+ network_interface_id = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
~ primary_network_interface_id = "eni-0f1ce5bdae258b015" -> (known after apply)
~ private_dns = "ip-172-31-61-141.ec2.internal" -> (known after apply)
~ private_ip = "172.31.61.141" -> (known after apply)
~ public_dns = "ec2-54-166-19-244.compute-1.amazonaws.com" -> (known after apply)
~ public_ip = "54.166.19.244" -> (known after apply)
~ security_groups = [
- "default",
] -> (known after apply)
source_dest_check = true
~ subnet_id = "subnet-1facdf35" -> (known after apply)

~ tenancy = "default" -> (known after apply)
~ volume_tags = {} -> (known after apply)
~ vpc_security_group_ids = [
- "sg-5255f429",
] -> (known after apply)
- credit_specification {
- cpu_credits = "standard" -> null
}
+ ebs_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ snapshot_id = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
+ ephemeral_block_device {
+ device_name = (known after apply)
+ no_device = (known after apply)
+ virtual_name = (known after apply)
}
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
~ root_block_device {
~ delete_on_termination = true -> (known after apply)
~ iops = 100 -> (known after apply)
~ volume_id = "vol-0079e485d9e28a8e5" -> (known after apply)
~ volume_size = 8 -> (known after apply)
~ volume_type = "gp2" -> (known after apply)
}
}
Plan: 1 to add, 0 to change, 1 to destroy.

NEW QUESTION: 88
Why is it a good idea to declare the required version of a provider in a Terraform configuration file?
1. terraform

2. {
3. required_providers
4. {
5. aws = "~> 1.0"
6. }
7. }

A. To remove older versions of the provider.
B. To match the version number of your application being deployed via Terraform.
C. Providers are released on a separate schedule from Terraform itself; therefore a newer version could introduce breaking changes.
D. To ensure that the provider version matches the version of Terraform you are using.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 89
Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)
A. CVS Version Control
B. GitHub
C. Bitbucket Cloud
D. Azure DevOps Server
E. GitHub Enterprise
Explanation
Terraform Cloud supports the following VCS providers:
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github-enterprise.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-com.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html
https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers
Answer: B,C,D,E (LEAVE A REPLY)

NEW QUESTION: 90
By default, where does Terraform store its state file?
A. Amazon S3 bucket
B. shared directory
C. remotely using Terraform Cloud
D. current working directory

Answer: D (LEAVE A REPLY)
By default, the state file is stored in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

NEW QUESTION: 91
You just scaled your VM infrastructure and realized you set the count variable to the wrong value. You correct the value and save your change.
What do you do next to make your infrastructure match your configuration?
A. Inspect all Terraform outputs to make sure they are correct
B. Run an apply and confirm the planned changes
C. Inspect your Terraform state because you want to change it
D. Reinitialize because your configuration has changed
Answer: A (LEAVE A REPLY)

NEW QUESTION: 92
Which argument(s) is (are) required when declaring a Terraform variable?
A. type
B. default
C. description
D. All of the above
E. None of the above
Answer: B (LEAVE A REPLY)
The variable declaration can also include a default argument.

NEW QUESTION: 93
Examine the following Terraform configuration, which uses the data source for an AWS AMI.
What value should you enter for the ami argument in the AWS instance resource?

A. aws_ami.ubuntu

B. data.aws_ami.ubuntu

C. data.aws_ami.ubuntu.id

D. aws_ami.ubuntu.id

Answer: C (LEAVE A REPLY)

Explanation

resource "aws_instance" "web" {

ami= data.aws_ami.ubuntu.id

Reference: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance

NEW QUESTION: 94

What is the command you can use to set an environment variable named "var1"of type String?

A. export TF_VAR_VAR1

B. set TF_VAR_var1

C. variable "var1" { type = "string"}

D. export TF_VAR_var1

Answer: D (LEAVE A REPLY)

Explanation

The environment variable must be in the format TF_VAR_name, so for the quetion: TF_VAR_var1 is the correct choice.

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

NEW QUESTION: 95

Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements.

(select three)

A. The module must be PCI/HIPPA compliant.

B. Module repositories must use this three-part name format, terraform-- .

C. The registry uses tags to identify module versions.

D. Release tag names must be for the format x.y.z, and can optionally be prefixed with a v .

E. The module must be on GitHub and must be a public repo.

Answer: C,D,E (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/registry/modules/publish.html#requirements

NEW QUESTION: 96
Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?
A. False
B. True
Answer: B (LEAVE A REPLY)
For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.
For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
https://www.terraform.io/docs/state/purpose.html

NEW QUESTION: 97
Which of these is the best practice to protect sensitive values in state files?
A. Blockchain
B. Secure Sockets Layer (SSL)
C. Enhanced remote backends
D. Signed Terraform providers
Answer: C (LEAVE A REPLY)
Use of remote backends and especially the availability of Terraform Cloud, there are now a variety of backends that will encrypt state at rest and will not store the state in cleartext on machines running. Reference: https://www.terraform.io/docs/extend/best-practices/sensitive-state.html

NEW QUESTION: 98
Setting the TF_LOG environment variable to DEBUG causes debug messages to be logged into syslog.
A. True
B. False
Answer: A (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 99
lookup retrieves the value of a single element from which of the below data type?
A. map
B. set

C. string
D. list
Answer: A (LEAVE A REPLY)
https://www.terraform.io/docs/configuration/functions/lookup.html

NEW QUESTION: 100
You have deployed a new webapp with a public IP address on a clod provider. However, you did not create any outputs for your code.
What is the best method to quickly find the IP address of the resource you deployed?
A. Run terraform destroy then terraform apply and look for the IP address in stdout
B. Run terraform state list to find the name of the resource, then terraform state show to find the attributes including public IP address
C. In a new folder, use the terraform_remote_state data source to load in the state file, then write an output for each resource that you find the state file
D. Run terraform output ip_address to view the result
Answer: (SHOW ANSWER)

NEW QUESTION: 101
Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.
A. True
B. False
Answer: A (LEAVE A REPLY)

NEW QUESTION: 102
One remote backend configuration always maps to a single remote workspace.
A. True
B. False
Answer: A (LEAVE A REPLY)
Explanation/Reference: https://www.terraform.io/docs/language/settings/backends/remote.html

NEW QUESTION: 103
What is the purpose of using the local-exec provisioner? (Select Two)
A. To invoke a local executable.
B. Executes a command on the resource to invoke an update to the Terraform state.
C. To execute one or more commands on the machine running Terraform.
D. Ensures that the resource is only executed in the local infrastructure where Terraform is deployed.
Answer: A,C (LEAVE A REPLY)
Explanation
The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource.

Note that even though the resource will be fully created when the provisioner is run, there is no guarantee that it will be in an operable state - for example system services such as sshd may not be started yet on compute resources.
Example usage
resource "aws_instance" "web" {
# ...
provisioner "local-exec" {
command = "echo ${aws_instance.web.private_ip} >> private_ips.txt"
}
}
Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives.
https://www.terraform.io/docs/provisioners/local-exec.html

NEW QUESTION: 104
Select two answers to complete the following sentence: Before a new provider can be used, it must be _____ and _____.
A. approved by HashiCorp
B. uploaded to source control
C. declared in the configuration
D. initialized
Answer: C,D (LEAVE A REPLY)
Explanation
Each time a new provider is added to configuration -- either explicitly via a provider block or by adding a resource from that provider -- Terraform must initialize the provider before it can be used. Initialization downloads and installs the provider's plugin so that it can later be executed.

NEW QUESTION: 105
True or False. The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. If drift is detected between the real-world infrastructure and the last known-state, it will modify the infrastructure to correct the drift.
A. False
B. True
Answer: A (LEAVE A REPLY)
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION: 106
Which of the following state management command allow you to retrieve a list of resources that are part of the state file?
A. terraform state list
B. terraform state view
C. terraform view
D. terraform list
Answer: A (LEAVE A REPLY)
Explanation

The terraform state list command is used to list resources within a Terraform state.
Usage: terraform state list [options] [address...]
The command will list all resources in the state file matching the given addresses (if any). If no addresses are given, all resources are listed.
https://www.terraform.io/docs/commands/state/list.html

NEW QUESTION: 107
You can migrate the Terraform backend but only if there are no resources currently being managed.
A. False
B. True
Answer: A (LEAVE A REPLY)
Explanation
If you need to migrate to another backend, such as Terraform Cloud, so you can continue managing it. By migrating your Terraform state, you can hand off infrastructure without de-provisioning anything.
https://www.terraform.io/docs/cloud/migrate/index.html

NEW QUESTION: 108
Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?
A. False
B. True
Answer: (SHOW ANSWER)
Terraform module need not be public and open-source. Module can be placed in -
* Local paths
* Terraform Registry
* GitHub
* Bitbucket
* Generic Git, Mercurial repositories
* HTTP URLs
* S3 buckets
* GCS buckets
https://www.terraform.io/docs/modules/sources.html

NEW QUESTION: 109
Which option can not be used to keep secrets out of Terraform configuration files?
A. A Terraform provider
B. Environment variables
C. A -var flag
D. secure string
Answer: C (LEAVE A REPLY)
Reference: https://secrethub.io/blog/secret-management-for-terraform/

NEW QUESTION: 110
Your team uses terraform OSS . You have created a number of resuable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option/way to do that?
A. Terraform modules cannot be shared in OSS version . Each developer needs to maintain their own modules and leverage them in the main tf file.
B. Upload your modules with proper versioning in the terraform public module registry . Terraform OSS is directly integrated with the public module registry , and can reference the modules from the code in the main tf file.
C. Terraform module sharing is only available in Enterprise version via terraform private module registry , so no way to enable it in OSS version.
D. Store your modules in a NAS/ shared file server , and ask your team members to directly reference the code from there. This is the only viable option in terraform OSS ,which is better than individually maintaining module versions for every developer.
Answer: (SHOW ANSWER)
Software development encourages code reuse through reusable artifacts, such as libraries, packages and modules. Most programming languages enable developers to package and publish these reusable components and make them available on a registry or feed. For example, Python has Python Package Index and PowerShell has PowerShell Gallery.
For Terraform users, the Terraform Registry enables the distribution of Terraform modules, which are reusable configurations. The Terraform Registry acts as a centralized repository for module sharing, making modules easier to discover and reuse.
The Registry is available in two variants:
* Public Registry houses official Terraform providers -- which are services that interact with an API to expose and manage a specific resource -- and community-contributed modules.
* Private Registry is available as part of the Terraform Cloud, and can host modules internally within an organization.
https://www.terraform.io/docs/registry/index.html

NEW QUESTION: 111
Terraform-specific settings and behaviors are declared in which configuration block type?
A. provider
B. terraform
C. resource
D. data
Answer: (SHOW ANSWER)
The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.

Example
terraform {
required_version = "> 0.12.0"
}
https://www.terraform.io/docs/configuration/terraform.html

NEW QUESTION: 112
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. What command will do this?
A. terraform taint
B. terraform apply
C. terraform graph
D. terraform refresh
Answer: A (LEAVE A REPLY)
The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.
Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.
This example will taint a single resource:
$ terraform taint aws_security_group.allow_all
The resource aws_security_group.allow_all in the module root has been marked as tainted.
https://www.terraform.io/docs/commands/taint.html

NEW QUESTION: 113
In Terraform Enterprise, a workspace can be mapped to how many VCS repos?
A. 5
B. 2
C. 3
D. 1
Answer: D (LEAVE A REPLY)
A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.
https://www.terraform.io/docs/cloud/workspaces/vcs.html

NEW QUESTION: 114

Which of the following best describes a Terraform provider?
A. A plugin that Terraform uses to translate the API interactions with the service or provider.
B. Serves as a parameter for a Terraform module that allows a module to be customized.
C. Describes an infrastructure object, such as a virtual network, compute instance, or other components.
D. A container for multiple resources that are used together.
Answer: A (LEAVE A REPLY)
Explanation
A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g. Terraform Cloud, DNSimple, Cloudflare).
https://www.terraform.io/docs/providers/index.html

NEW QUESTION: 115
Talal is a DevOps engineer and he has deployed the production infrastructure using Terraform. He is using a very large configuration file to maintain and update the actual infrastructure. As the infrastructure have grown to a very complex and large, he has started experiencing slowness when he run runs terraform plan. What are the options for him to resolve this slowness?
A. Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.
B. Run terraform refresh every time before running terraform plan.
C. Break large configurations into several smaller configurations that can each be independently applied.
D. Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.
Answer: C,D (LEAVE A REPLY)
Explanation
For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
Although 'Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.' is a solution, but its not always recommended. Instead of using -target as a means to operate on isolated portions of very large configurations, prefer instead to break large configurations into several smaller configurations that can each be independently applied. Data sources can be used to access information about resources created in other configurations, allowing a complex system architecture to be broken down into more manageable parts that can be updated independently.
Option 'Run terraform refresh every time before running terraform plan.' and 'Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.' is not correct because in both the cases terraform will query every resources of the infrastructure.

NEW QUESTION: 116
What does terraform refresh modify?
A. Your cloud infrastructure
B. Your Terraform plan
C. Your state file

D. Your Terraform configuration
Answer: C (LEAVE A REPLY)

NEW QUESTION: 117
Which one of the following will run echo 0 and echo 1 on a newly created host?
A. provisioner "local-exec" { command = "echo 0"
command = "echo 1"
}
B. provisioner "remote-exec" {
inline = [
echo 0,
echo 1
]
}
C. provisioner "remote-exec" {
command = "${echo 0}"
command = "${echo 1}"
}
D. provisioner "remote-exec" {
inline = [
"echo 0",
"echo 1"
]
}
Answer: D (LEAVE A REPLY)
remote-exec Provisioner
Example usage
resource "aws_instance" "web" {
# ...
provisioner "remote-exec" {
inline = [
"puppet apply",
"consul join ${aws_instance.web.private_ip}",
]
}
}

NEW QUESTION: 118
Which of the following statements best describes the Terraform list(...) type?
A. a collection of values where each is identified by a string label.
B. a sequence of values identified by consecutive whole numbers starting with zero.

C. a collection of unique values that do not have any secondary identifiers or ordering.
D. a collection of named attributes that each have their own type.
Answer: (SHOW ANSWER)
A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.
https://www.terraform.io/docs/configuration/types.html#structural-types

NEW QUESTION: 119
Given the below resource configuration -
resource "aws_instance" "web" { # ... count = 4 }
What does the terraform resource address aws_instance.web refer to?
A. It refers to all 4 web instances , together , for further individual segregation , indexing is required , with a 0 based index.
B. It refers to the last web EC2 instance , as by default , if no index is provided , the last / N-1 index is used.
C. It refers to the first web EC2 instance out of the 4 ,as by default , if no index is provided , the first / 0th index is used.
D. The above will result in a syntax error , as it is not syntactically correct . Resources defined using count , can only be referenced using indexes.
Answer: A (LEAVE A REPLY)
A Resource Address is a string that references a specific resource in a larger infrastructure. An address is made up of two parts:
[module path][resource spec]
Module path:
A module path addresses a module within the tree of modules. It takes the form:
module.A.module.B.module.C...
Multiple modules in a path indicate nesting. If a module path is specified without a resource spec, the address applies to every resource within the module. If the module path is omitted, this addresses the root module.
Given a Terraform config that includes:
resource "aws_instance" "web" {
# ...
count = 4
}
An address like this:
aws_instance.web[3]
Refers to only the last instance in the config, and an address like this:
aws_instance.web
Refers to all four "web" instances.
https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION: 120
What is one disadvantage of using dynamic blocks in Terraform?
A. They cannot be used to loop through a list of values
B. Dynamic blocks can construct repeatable nested blocks
C. They make configuration harder to read and understand
D. Terraform will run more slowly

Answer: (SHOW ANSWER)
Explanation/Reference: https://github.com/hashicorp/terraform/issues/19291

NEW QUESTION: 121
Select the most accurate statement to describe the Terraform language from the following list.
A. Terraform is an immutable, declarative, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally JSON.
B. Terraform is a mutable, declarative, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
C. Terraform is an immutable, procedural, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
D. Terraform is a mutable, procedural, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally YAML.
Answer: (SHOW ANSWER)
Terraform is not a configuration management tool - https://www.terraform.io/intro/vs/chefpuppet.html Terraform is a declarative language - https://www.terraform.io/docs/configuration/index.html Terraform supports a syntax that is JSON compatible - https://www.terraform.io/docs/configuration/syntax-json.html Terraform is primarily designed on immutable infrastructure principles - https://www.hashicorp.com/resources/what-is-mutable-vs-immutable-infrastructure

NEW QUESTION: 122
What is the workflow for deploying new infrastructure with Terraform?
A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.
C. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure
Answer: C (LEAVE A REPLY)
changes, and terraform apply to create new infrastructure.
Reference:
+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&oq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new

+infrastructure.&aqs=chrome..69i57.556j0j7&sourceid=chrome&ie=UTF-8

NEW QUESTION: 123
In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to use what?
A. Input Parameters
B. Module
C. Regular Expressions
D. Output Value
Answer: A (LEAVE A REPLY)
Explanation
Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.
https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION: 124
Terraform variables and outputs that set the "description" argument will store that description in the state file.
A. True
B. False
Answer: (SHOW ANSWER)
Explanation/Reference: https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION: 125
Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?
A. Use data source aws_ami.
B. Create a map of region to ami id.
C. Create different configuration file for different region.
D. None of the above
Answer: (SHOW ANSWER)
https://www.terraform.io/docs/configuration/data-sources.html

NEW QUESTION: 126
By default, a defined provisioner is a creation-time provisioner.
A. True
B. False
Answer: (SHOW ANSWER)
Explanation
https://www.terraform.io/docs/provisioners/index.html

NEW QUESTION: 127

John wants to use two different regions to deploy two different EC2 instances. He has specified two provider blocks in his providers.tf file.
provider "aws" { region = "us-east-1" }
provider "aws" { region = "us-west-2" }
When he run terraform plan he encountered an error. How to fix this?
A. It can not be fixed
B. Use alias for region = "us-west-2"
C. Use another provider version
D. Use default keyword with region = "us-east-1"
Answer: B (LEAVE A REPLY)

NEW QUESTION: 128
You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.
What are the two things you must do to achieve this? (Choose two.)
A. Provision new VMs using Terraform with the same VM names
B. Use the terraform import command for the existing VMs
C. Write Terraform configuration for the existing VMs
D. Run the terraform import-gcp command
Answer: B,D (LEAVE A REPLY)
Explanation
The terraform import command is used to import existing infrastructure. Import existing Google Cloud resources into Terraform with Terraformer.
Reference: https://www.terraform.io/docs/cli/import/usage.html https://cloud.google.com/docs/terraform

NEW QUESTION: 129
Your team uses terraform OSS . You have created a number of resuable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option/way to do that?
A. Terraform modules cannot be shared in OSS version . Each developer needs to maintain their own modules and leverage them in the main tf file.
B. Upload your modules with proper versioning in the terraform public module registry . Terraform OSS is directly integrated with the public module registry , and can reference the modules from the code in the main tf file.
C. Terraform module sharing is only available in Enterprise version via terraform private module registry , so no way to enable it in OSS version.
D. Store your modules in a NAS/ shared file server , and ask your team members to directly reference the code from there. This is the only viable option in terraform OSS ,which is better than individually maintaining module versions for every developer.
Answer: (SHOW ANSWER)
Explanation
Software development encourages code reuse through reusable artifacts, such as libraries, packages and modules. Most programming languages enable developers to package and publish these reusable components and make them available on a registry or feed. For example, Python has Python Package Index and PowerShell has PowerShell Gallery.

For Terraform users, the Terraform Registry enables the distribution of Terraform modules, which are reusable configurations. The Terraform Registry acts as a centralized repository for module sharing, making modules easier to discover and reuse.
The Registry is available in two variants:
* Public Registry houses official Terraform providers -- which are services that interact with an API to expose and manage a specific resource -- and community-contributed modules.
* Private Registry is available as part of the Terraform Cloud, and can host modules internally within an organization.
https://www.terraform.io/docs/registry/index.html

NEW QUESTION: 130
A Terraform provider is not responsible for:
A. Exposing resources and data sources based on an API
B. Provisioning infrastructure in multiple clouds
C. Managing actions to take based on resource differences
D. Understanding API interactions with some service
Answer: C (LEAVE A REPLY)

NEW QUESTION: 131
What are some of the features of Terraform state? (select three)
A. increased performance
B. inspection of cloud resources
C. mapping configuration to real-world resources
D. determining the correct order to destroy resources
Answer: (SHOW ANSWER)

NEW QUESTION: 132
What command does Terraform require the first time you run it within a configuration directory?
A. terraform import
B. terraform init
C. terraform plan
D. terraform workspace
Answer: B (LEAVE A REPLY)
terraform init command is used to initialize a working directory containing Terraform configuration files.
Reference: https://www.terraform.io/docs/cli/commands/init.html

NEW QUESTION: 133
You need to specify a dependency manually. What resource meta-parameter can you use lo make sure Terraform respects thee dependency?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.
Answer:
dependson

NEW QUESTION: 134
Using the terraform state rm command against a resource will destroy it.
A. False
B. True
Answer: A (LEAVE A REPLY)

NEW QUESTION: 135
Your security team scanned some Terraform workspaces and found secrets stored in a plaintext in state files.
How can you protect sensitive data stored in Terraform state files?
A. Edit your state file to scrub out the sensitive data
B. Store the state in an encrypted backend
C. Always store your secrets in a secrets.tfvars file.
D. Delete the state file every time you run Terraform
Answer: B (LEAVE A REPLY)

NEW QUESTION: 136
True or False. The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. If drift is detected between the real-world infrastructure and the last known-state, it will modify the infrastructure to correct the drift.
A. False
B. True
Explanation
https://www.terraform.io/docs/commands/refresh.html
Answer: (SHOW ANSWER)

NEW QUESTION: 137
How would you be able to reference an attribute from the vsphere_datacenter data source for use with the argument within the vsprere_folder resource in the following configuration?

A. data.dc,id
B. data.vsphere_datacenter.dc.id
C. vsphere_datacenter.dc.id
D. data.vsphere_datacenter.dc
Answer: B (LEAVE A REPLY)

NEW QUESTION: 138
You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed?
A. Terraform can still determine the correct order for destruction from the state even when you delete one or more items from the configuration.
B. Those would be destroyed in the order in which they were written in the configuration file previously before you have deleted them from configuration file.
C. The resource will be destructed in random order as you have already deleted them from configuration.
D. You can not destroy resources by deleting them from configuration file and running plan and apply.
Answer: A (LEAVE A REPLY)
Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.
To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.

NEW QUESTION: 139
Multiple configurations for the same provider can be used in a single configuration file.
A. False
B. True
Answer: B (LEAVE A REPLY)
Explanation
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:
# The default provider configuration
provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region
provider "aws" {
alias = "west"
region = "us-west-2"
}
The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances

NEW QUESTION: 140
Your DevOps team is currently using the local backend for your Terraform configuration. You would like to move to a remote backend to begin storing the state file in a central location.
Which of the following backends would not work?
A. Git
B. Terraform Cloud
C. Amazon S3
D. Artifactory
Answer: (SHOW ANSWER)

NEW QUESTION: 141
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.
A. True
B. False
Answer: A (LEAVE A REPLY)

NEW QUESTION: 142
You have configured an Auto Scaling group in AWS to automatically scale the number of instances behind a load balancer based on the instances CPU utilization. The instances are configured using a Launch Configuration. You have observed that the Auto Scaling group doesn't successfully scale when you apply changes that require replacing the Launch Configuration. Why is this happening?
A. You need to configure an explicit dependency for the Auto Scaling group using the depends_on meta-parameter.
B. You need to configure an explicit dependency for the Launch Configuration using the depends_on meta-parameter.
C. You need to configure the Auto Scaling group's create_before_destroy meta-parameter.
D. You need to configure the Launch Configuration's create_before_destroy meta-parameter.
Answer: D (LEAVE A REPLY)

Explanation
https://www.terraform.io/docs/providers/aws/r/launch_configuration.html#using-withautoscaling-groups

NEW QUESTION: 143
You have created two workspaces PROD and DEV. You have switched to DEV and provisioned DEV infrastructure from this workspace. Where is your state file stored?
A. terraform.d
B. terraform.tfstate
C. terraform.tfstate.DEV
D. terraform.tfstate.d
Answer: D (LEAVE A REPLY)
Explanation
Terraform stores the workspace states in a directory called terraform.tfstate.d. This directory should be treated similarly to default workspace state file terraform.tfstate main.tf provider.tf terraform.tfstate.d DEV terraform.tfstate # DEV workspace state file PROD terraform.tfstate # PROD workspace state file terraform.tfvars # Default workspace state file variables.tf

NEW QUESTION: 144
Environment variables can be used to set variables. The environment variables must be in the format "____"_<variablename>. Select the correct prefix string from the following list.
A. TF_CLI_ARGS
B. TF_VAR
C. TF_VAR_
D. TF_VAR_ENV
Answer: C (LEAVE A REPLY)
Environment variables can be used to set variables. The environment variables must be in the format TF_VAR_name and this will be checked last for a value. For example:
export TF_VAR_region=us-west-1
export TF_VAR_ami=ami-049d8641
export TF_VAR_alist='[1,2,3]'
export TF_VAR_amap='{ foo = "bar", baz = "qux" }'
https://www.terraform.io/docs/commands/environment-variables.html

NEW QUESTION: 145
A single terraform resource file that defines an aws_instance resource can simply be renamed to vsphere_virtual_machine in order to switch cloud providers.
A. True
B. False
Answer: B (LEAVE A REPLY)
Explanation
Every provider has its own required and allowed declarations none of which match between cloud providers.

NEW QUESTION: 146
Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?
A. False
B. True
Answer: (SHOW ANSWER)
Explanation
For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.
For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
https://www.terraform.io/docs/state/purpose.html

NEW QUESTION: 147
What are the benefits of using Infrastructure as Code? (select five)
A. Infrastructure as Code provides configuration consistency and standardization among deployments
B. Infrastructure as Code allows a user to turn a manual task into a simple, automated deployment (Correct) Explanation If you are new to infrastructure as code as a concept, it is the process of managing infrastructure in a file or files rather than manually configuring resources in a user interface.
A resource in this instance is any piece of infrastructure in a given environment, such as a virtual machine, security group, network interface, etc. At a high level, Terraform allows operators to use HCL to author files containing definitions of their desired resources on almost any provider (AWS, GCP, GitHub, Docker, etc) and automates the creation of those resources at the time of application.
C. Infrastructure as Code is easily repeatable, allowing the user to reuse code to deploy similar, yet different resources
D. Infrastructure as Code is relatively simple to learn and write, regardless of a user's prior experience with developing code
E. Infrastructure as Code gives the user the ability to recreate an application's infrastructure for disaster recovery scenarios
F. Infrastructure as Code easily replaces development languages such as Go and .Net for application development
Answer: (SHOW ANSWER)

NEW QUESTION: 148
In contrast to Terraform Open Source, when working with Terraform Enterprise and Cloud Workspaces, conceptually you could think about them as completely separate working directories.
A. False
B. True
Answer: (SHOW ANSWER)

NEW QUESTION: 149

John is writing a module and within the module, there are multiple places where he has to use the same conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration,. What is a better approach to dealing with this?

A. Local Values

B. Expressions

C. Functions

D. Variables

Answer: A (LEAVE A REPLY)

Explanation

A local value assigns a name to an expression, allowing it to be used multiple times within a module without repeating it.

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

NEW QUESTION: 150

When should you use the force-unlock command?

A. You see a status message that you cannot acquire the lock

B. You have a high priority change

C. Automatic unlocking failed

D. Your apply failed due to a state lock

Answer: (SHOW ANSWER)

Manually unlock the state for the defined configuration.

NEW QUESTION: 151

By default, provisioners that fail will also cause the Terraform apply itself to error. How can you change this default behavior within a provisioner?

A. provisioner "local-exec" { on_failure = "next" }

B. provisioner "local-exec" { when = "failure" terraform apply }

C. provisioner "local-exec" { on_failure = "continue" }

D. provisioner "local-exec" { on_failure = continue }

Answer: C (LEAVE A REPLY)

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

NEW QUESTION: 152

Talal is a DevOps engineer and he has deployed the production infrastructure using Terraform. He is using a very large configuration file to maintain and update the actual infrastructure. As the infrastructure have grown to a very complex and large, he has started experiencing slowness when he run runs terraform plan. What are the options for him to resolve this slowness?

A. Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.

B. Run terraform refresh every time before running terraform plan.

C. Break large configurations into several smaller configurations that can each be independently applied.

D. Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.

Answer: (SHOW ANSWER)

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.

Although 'Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.' is a solution, but its not always recommended. Instead of using -target as a means to operate on isolated portions of very large configurations, prefer instead to break large configurations into several smaller configurations that can each be independently applied. Data sources can be used to access information about resources created in other configurations, allowing a complex system architecture to be broken down into more manageable parts that can be updated independently.

Option 'Run terraform refresh every time before running terraform plan.' and 'Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.' is not correct because in both the cases terraform will query every resources of the infrastructure.

NEW QUESTION: 153

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

A. format

B. replace

C. join

D. tostring

Explanation

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

Answer: (SHOW ANSWER)

NEW QUESTION: 154

In Terraform 0.13 and above, outside of the required_providers block, Terraform configurations always refer to providers by their local names.

A. True

B. False

Answer: A (LEAVE A REPLY)

Explanation

Outside of the required_providers block, Terraform configurations always refer to providers by their local names.

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

NEW QUESTION: 155
Which of the following is not a key principle of infrastructure as code?
A. Idempotence
B. Self-describing infrastructure
C. Golden images
D. Versioned infrastructure
Answer: B,C,D (LEAVE A REPLY)

NEW QUESTION: 156
What type of block is used to construct a collection of nested configuration blocks?
A. dynamic
B. nesting
C. for_each
D. repeated
Answer: A (LEAVE A REPLY)

NEW QUESTION: 157
If you enable TF_LOG = DEBUG, the log will be stored in syslog.log file in the currect directory.
A. False
B. True
Answer: A (LEAVE A REPLY)
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 158
What features does the hosted service Terraform Cloud provide? (Choose two.)
A. Automated infrastructure deployment visualization
B. Automatic backups
C. Remote state storage
D. A web-based user interface (UI)
Answer: B,C (LEAVE A REPLY)
Reference:
https://www.terraform.io/docs/enterprise/admin/automated-recovery.html
https://www.terraform.io/docs/language/state/remote.html

NEW QUESTION: 159
Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements. (select three)
A. The module must be PCI/HIPPA compliant.
B. Module repositories must use this three-part name format, terraform-- .
C. The registry uses tags to identify module versions.

D. Release tag names must be for the format x.y.z, and can optionally be prefixed with a v .
E. The module must be on GitHub and must be a public repo.
Answer: C,D,E (LEAVE A REPLY)
https://www.terraform.io/docs/registry/modules/publish.html#requirements

NEW QUESTION: 160
Which of the following is not true of Terraform providers?
A. Providers can be written by individuals
B. Providers can be maintained by a community of users
C. Some providers are maintained by HashiCorp
D. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers
E. None of the above
Answer: (SHOW ANSWER)
Reference: https://jayendrapatil.com/terraform-cheat-sheet/#Terraform_Read_and_write_configuration

NEW QUESTION: 161
What are the benefits of using Infrastructure as Code? (select five)
A. Infrastructure as Code is relatively simple to learn and write, regardless of a user's prior experience with developing code
B. Infrastructure as Code provides configuration consistency and standardization among deployments
C. Infrastructure as Code is easily repeatable, allowing the user to reuse code to deploy similar, yet different resources
D. Infrastructure as Code gives the user the ability to recreate an application's infrastructure for disaster recovery scenarios
E. Infrastructure as Code easily replaces development languages such as Go and .Net for application development
F. Infrastructure as Code allows a user to turn a manual task into a simple, automated deployment (Correct)
Answer: A,C,D,F (LEAVE A REPLY)
If you are new to infrastructure as code as a concept, it is the process of managing infrastructure in a file or files rather than manually configuring resources in a user interface.
A resource in this instance is any piece of infrastructure in a given environment, such as a virtual machine, security group, network interface, etc. At a high level, Terraform allows operators to use HCL to author files containing definitions of their desired resources on almost any provider (AWS, GCP, GitHub, Docker, etc) and automates the creation of those resources at the time of application.

NEW QUESTION: 162
Terraform Enterprise currently supports running under which the following operating systems?
A. Ubuntu
B. Amazon Linux
C. Debian
D. CentOS
E. Red Hat Enterprise Linux
F. Oracle Linux
Answer: (SHOW ANSWER)
Explanation

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

NEW QUESTION: 163
Which of the following are string functions? Select three
A. tostring
B. tonumber
C. Chomp
D. format
E. join
Answer: C,D,E (LEAVE A REPLY)
tonumber and tostring are Type Conversion function
https://www.terraform.io/docs/configuration/functions.html

NEW QUESTION: 164
Terraform providers are always installed from the Internet.
A. True
B. False
Answer: B (LEAVE A REPLY)
Explanation
Terraform configurations must declare which providers they require, so that Terraform can install and use them.
Reference: https://www.terraform.io/docs/language/providers/configuration.html

NEW QUESTION: 165
You have written a terraform IaC script which was working till yesterday , but is giving some vague error from today , which you are unable to understand . You want more detailed logs that could potentially help you troubleshoot the issue , and understand the root cause. What can you do to enable this setting? Please note , you are using terraform OSS.
A. Terraform OSS can push all its logs to a syslog endpoint. As such, you have to set up the syslog sink, and enable TF_LOG_PATH env variable to the syslog endpoint and all logs will automatically start streaming.
B. Detailed logs are not available in terraform OSS, except the crash message. You need to upgrade to terraform enterprise for this point.
C. Enable the TF_LOG_PATH to the log sink file location, and logging output will automatically be stored there.

D. Enable TF_LOG to the log level DEBUG, and then set TF_LOG_PATH to the log sink file location. Terraform debug logs will be dumped to the sink path, even in terraform OSS.
Answer: D (LEAVE A REPLY)
Terraform has detailed logs which can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr.
You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.
TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name.
To persist logged output you can set TF_LOG_PATH in order to force the log to always be appended to a specific file when logging is enabled. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

NEW QUESTION: 166
Which of the below are paid features of Terraform Cloud?
A. Full API Coverage
B. Secure variable Storage
C. Roles/ Team management
D. Cost Estimation
E. Private Module Registry
F. Sentinel policies
Answer: (SHOW ANSWER)
Explanation
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION: 167
Which one is the right way to import a local module names consul?
A. module "consul" { source = "consul"}
B. module "consul" { source = "./consul"}
C. module "consul" { source = "../consul"}
D. module "consul" { source = "module/consul"}
Answer: B,C (LEAVE A REPLY)
A local path must begin with either ./ or ../ to indicate that a local path is intended, to distinguish from a module registry address.
module "consul" {
source = "./consul"
}

NEW QUESTION: 168
Refer below code where pessimistic constraint operator has been used to specify a version of a provider.
terraform { required_providers { aws = "~> 1.1.0" }}
Which of the following options are valid provider versions that satisfy the above constraint. (select two)
A. 1.1.1
B. 1.2.9
C. 1.1.8
D. 1.2.0
Answer: A,C (LEAVE A REPLY)
Pessimistic constraint operator, constraining both the oldest and newest version allowed. For example, ~> 0.9 is equivalent to >= 0.9, < 1.0, and ~> 0.8.4, is equivalent to >= 0.8.4, < 0.9

NEW QUESTION: 169
You should store secret data in the same version control repository as your Terraform configuration.
A. True
B. False
Answer: (SHOW ANSWER)
Reference: https://blog.gruntwork.io/a-comprehensive-guide-to-managing-secrets-in-your-terraform-code-1d586955ace1

NEW QUESTION: 170
Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 ...) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region. What should you do? Your provider setting is as below -
The default provider configuration provider "aws" { region = "us-east-1" }
A. No way to enable this via a single script . Write 2 different scripts with different default providers in the 2 scripts , one for us-east , another for us-west.
B. Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.
C. Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).
D. Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.
Answer: (SHOW ANSWER)
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

```
# The default provider configuration
provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region
provider "aws" {
alias = "west"
region = "us-west-2"
}
```
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION: 171
What does the default "local" Terraform backend store?
A. tfplan files
B. Terraform binary
C. Provider plugins
D. State file
Answer: D (LEAVE A REPLY)
Explanation
The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.
Reference: https://www.terraform.io/docs/language/settings/backends/local.html

NEW QUESTION: 172
lookup retrieves the value of a single element from which of the below data type?
A. map
B. set
C. string
D. list
Answer: A (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/configuration/functions/lookup.html

NEW QUESTION: 173
From the answers below, select the advantages of using Infrastructure as Code.
A. Provide a codified workflow to develop customer-facing applications.
B. Safely test modifications using a "dry run" before applying any actual changes.
C. Easily integrate with application workflows (GitLab Actions, Azure DevOps, CI/CD tools).
D. Easily change and update existing infrastructure.
E. Provide reusable modules for easy sharing and collaboration.
Answer: B,C,D,E (LEAVE A REPLY)
Explanation

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

NEW QUESTION: 174
You can migrate the Terraform backend but only if there are no resources currently being managed.
A. False
B. True
Answer: A (LEAVE A REPLY)
If you need to migrate to another backend, such as Terraform Cloud, so you can continue managing it. By migrating your Terraform state, you can hand off infrastructure without de-provisioning anything.
https://www.terraform.io/docs/cloud/migrate/index.html

NEW QUESTION: 175
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
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: C (LEAVE A REPLY)
* Indent two spaces for each nesting level.
* When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs.
Reference: https://www.terraform.io/docs/language/syntax/style.html

NEW QUESTION: 176
If a module uses a local variable, you can expose that value with a terraform output.
A. True
B. False
Answer: (SHOW ANSWER)
Output values are like function return values.

NEW QUESTION: 177
Which of the following clouds does not have a provider maintained HashiCorp?
A. IBM Cloud
B. DigitalOcean
C. OpenStack
D. AWS
Answer: A (LEAVE A REPLY)

Explanation
IBM Cloud does not have a provider maintained by HashiCorp, although IBM Cloud does maintain their own Terraform provider.
https://www.terraform.io/docs/providers/index.html

NEW QUESTION: 178
A user runs terraform init on their RHEL based server and per the output, two provider plugins are downloaded: $ terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...
- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...
:
Terraform has been successfully initialized! Where are these plugins downloaded to?
A. The .terraform.d directory in the directory terraform init was executed in.
B. The .terraform/plugins directory in the directory terraform init was executed in.
C. The .terraform.plugins directory in the directory terraform init was executed in.
D. /etc/terraform/plugins
Answer: (SHOW ANSWER)

NEW QUESTION: 179
When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)
1. terraform
2. {
3. required_providers
4. {
5. aws = "~> 1.2.0"
6. }
7. }
A. 1.3.1
B. 1.2.3
C. 1.2.9
D. 1.3.0
Answer: B,C (LEAVE A REPLY)
As your Terraform usage becomes more advanced, there are some cases where you may need to modify the Terraform state. Rather than modify the state directly, the terraform state commands can be used in many cases instead. This command is a nested subcommand, meaning that it has further subcommands.
https://www.terraform.io/docs/commands/state/index.html

NEW QUESTION: 180

Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. What command will do this?

A. terraform taint

B. terraform apply

C. terraform graph

D. terraform refresh

Answer: A (LEAVE A REPLY)

Explanation

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.

This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.

Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.

Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.

This example will taint a single resource:

$ terraform taint aws_security_group.allow_all

The resource aws_security_group.allow_all in the module root has been marked as tainted.

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

NEW QUESTION: 181

Workspaces in Terraform provides similar functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

A. True

B. False

Answer: B (LEAVE A REPLY)

https://www.terraform.io/docs/cloud/migrate/workspaces.html

Workspaces, managed with the terraform workspace command, aren't the same thing as Terraform Cloud's workspaces. Terraform Cloud workspaces act more like completely separate working directories; CLI workspaces are just alternate state files.

NEW QUESTION: 182
Please identify the offerings which are unique to Terraform Enterprise, and not available in either Terraform OSS, or Terraform Cloud. Select four.
A. Audit Logs
B. Private Network Connectivity
C. VCS Integration
D. Sentinel
E. Clustering
Answer: (SHOW ANSWER)
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION: 183
A data block requests that Terraform read from a given data source and export the result under the given local name.
A. False
B. True
Answer: B (LEAVE A REPLY)

NEW QUESTION: 184
You want terraform plan and terraform apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?
A. Local Backends.
B. Terraform Backends.
C. This can be done using any of the local or remote backends.
D. Remote Backends.
Answer: D (LEAVE A REPLY)
When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.
Terraform Cloud can also be used with local operations, in which case only state is stored in the Terraform Cloud backend.
https://www.terraform.io/docs/backends/types/remote.html

NEW QUESTION: 185
You have multiple developers working on a terraform project (using terraform OSS), and have saved the terraform state in a remote S3 bucket . However ,team is intermittently experiencing inconsistencies in the provisioned infrastructure / failure in the code . You have traced this problem to simultaneous/concurrent runs of terraform apply command for 2/more developers . What can you do to fix this problem?
A. Use terraform workspaces feature, this will fix this problem by default , as every developer will have their own state file , and terraform will merge them on server side on its own.
B. Structure your team in such a way that only one individual will run terraform apply , everyone will just make changes and share with him. Then there will be no chance of any inconsistencies.

C. Stop using remote state , and store the developer tfstate in their own machine . Once a day , all developers should sit together and merge the state files manually , to avoid any inconsistencies.
D. Enable terraform state locking for the S3 backend using DynamoDB table. This prevents others from acquiring the lock and potentially corrupting your state.
Answer: D (LEAVE A REPLY)
S3 backend support state locking using DynamoDB.
https://www.terraform.io/docs/state/locking.html

NEW QUESTION: 186
Which of the following clouds does not have a provider maintained HashiCorp?
A. IBM Cloud
B. DigitalOcean
C. OpenStack
D. AWS
Answer: A (LEAVE A REPLY)
IBM Cloud does not have a provider maintained by HashiCorp, although IBM Cloud does maintain their own Terraform provider.
https://www.terraform.io/docs/providers/index.html

NEW QUESTION: 187
Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.
A. Set TF_LOG = DEBUG
B. Set VAR_TF = TRACE
C. Set TF_LOG = TRACE
D. Set VAR_TF_LOG = TRACE
Answer: (SHOW ANSWER)
Explanation
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 188
Which of the below terraform commands do not run terraform refresh implicitly before taking actual action of the command?
A. terraform apply
B. terraform destroy
C. terraform init
D. terraform import
E. terraform plan
Answer: C,D (LEAVE A REPLY)
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION: 189

Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.
What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?
A. Submit a ticket to AWS and ask them to export the state of all existing resources and use terraform import to import them into the state file.
B. Delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward.
C. Resources that are manually deployed in the AWS console cannot be imported by Terraform.
D. Using terraform import, import the existing infrastructure into your Terraform state.
Answer: (SHOW ANSWER)
Explanation
Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.
This is a great way to slowly transition infrastructure to Terraform.
The terraform import command is used to import existing infrastructure.
To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.
Example:
resource "aws_instance" "import_example" {
# ...instance configuration...
}
Now terraform import can be run to attach an existing instance to this resource configuration.
$ terraform import aws_instance.import_example i-03efafa258104165f
aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f)
aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION: 190
Which of the below configuration file formats are supported by Terraform? (Select TWO)
A. Node
B. JSON
C. Go
D. YAML
E. HCL
Answer: B,E (LEAVE A REPLY)
Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations.

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

NEW QUESTION: 191
When Terraform needs to be installed in a location where it does not have internet access to download the installer and upgrades, the installation is generally known as to be _____.
A. a private install
B. disconnected
C. air-gapped
D. non-traditional
Answer: (SHOW ANSWER)
A Terraform Enterprise install that is provisioned on a network that does not have Internet access is generally known as an air-gapped install. These types of installs require you to pull updates, providers, etc. from external sources vs. being able to download them directly.

NEW QUESTION: 192
How is terraform import run?
A. As a part of terraform plan
B. By an explicit call
C. As a part of terraform init
D. As a part of terraform refresh
E. All of the above
Answer: A (LEAVE A REPLY)

NEW QUESTION: 193
If you enable TF_LOG = DEBUG, the log will be stored in syslog.log file in the currect directory.
A. False
B. True
Answer: (SHOW ANSWER)
Explanation
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 194
Complete the following sentence:
The terraform state command can be used to ____
A. view state
B. There is no such command
Explanation
https://www.terraform.io/docs/commands/state/index.html
C. modify state
D. refresh state
Answer: C (LEAVE A REPLY)

NEW QUESTION: 195
You have to initialize a Terraform backend before it can be configured.
A. False
B. True
Answer: (SHOW ANSWER)

NEW QUESTION: 196
You have been given requirements to create a security group for a new application. Since your organization standardizes on Terraform, you want to add this new security group with the fewest number of lines of code.
What feature could you use to iterate over a list of required tcp ports to add to the new security group?
A. dynamic backend
B. splat expression
C. terraform import
D. dynamic block
Answer: D (LEAVE A REPLY)
Explanation
A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value and generates a nested block for each element of that complex value.
https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks

NEW QUESTION: 197
Select two answers to complete the following sentence: Before a new provider can be used, it must be _____ and _____.
A. approved by HashiCorp
B. uploaded to source control
C. declared in the configuration
D. initialized
Answer: (SHOW ANSWER)
Each time a new provider is added to configuration -- either explicitly via a provider block or by adding a resource from that provider -- Terraform must initialize the provider before it can be used. Initialization downloads and installs the provider's plugin so that it can later be executed.

NEW QUESTION: 198

You have never used Terraform before and would like to test it out using a shared team account for a cloud provider. The shared team account already contains 15 virtual machines (VM). You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully.
What should you do to delete the newly-created VM with Terraform?
A. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.
B. The Terraform state file contains all 16 VMs in the team account. Execute terraform destroy and select the newly-created VM.
C. The Terraform state file only contains the one new VM. Execute terraform destroy.
D. Delete the Terraform state file and execute Terraform apply.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 199
You have provisioned some aws resources in your test environment through Terraform for a POC work. After the POC, now you want to destroy the resources but before destroying them you want to check what resources will be getting destroyed through terraform. what are the options of doing that? (Select TWO)
A. Use terraform destroy command
B. This is not possible
C. Use terraform plan command
D. Use terraform plan -destroy command.
Answer: A,D (LEAVE A REPLY)
Explanation
https://learn.hashicorp.com/terraform/getting-started/destroy

NEW QUESTION: 200
You're building a CI/CD (continuous integration/ continuous delivery) pipeline and need to inject sensitive variables into your Terraform run.
How can you do this safely?
A. Store the sensitive variables as plain text in a source code repository
B. Pass variables to Terraform with a -var flag
C. Store the sensitive variables in a secure_vars.tf file
D. Copy the sensitive variables into your Terraform code
Answer: (SHOW ANSWER)