HashiCorp.TA-002-P.v2022-05-14.q184

Exam Code:	TA-002-P
Exam Name:	HashiCorp Certified: Terraform Associate
Certification Provider:	HashiCorp
Free Question Number:	184
Version:	v2022-05-14
# of views:	3053
# of Questions views:	1840
https://www.dumpsdb.com/dumps/HashiCorp/TA-002-P/HashiCorp.TA-002-P.v2022-05-14.q184	

NEW QUESTION: 1
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: (SHOW ANSWER)
Explanation
A local path must begin with either ./ or ../ to indicate that a local path is intended, to distinguish from a module registry address.
module "consul" {
source = "./consul"
}

NEW QUESTION: 2
When running the command terraform taint against a managed resource you want to force recreation upon, Terraform will immediately destroy and recreate the resource.
A. True
B. False
Answer: (SHOW ANSWER)

NEW QUESTION: 3

resource "aws_s3_bucket" "example" { bucket = "my-test-s3-terraform-bucket" ...} resource "aws_iam_role" "test_role" { name = "test_role" ...} Due to the way that the application code is written , the s3 bucket must be created before the test role is created , otherwise there will be a problem. How can you ensure that?

A. This will already be taken care of by terraform native implicit dependency. Nothing else needs to be done from your end.

B. Add explicit dependency using depends_on . This will ensure the correct order of resource creation.

C. Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.

D. This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.

Answer: B (LEAVE A REPLY)

Use the depends_on meta-argument to handle hidden resource dependencies that Terraform can't automatically infer.

Explicitly specifying a dependency is only necessary when a resource relies on some other resource's behavior but doesn't access any of that resource's data in its arguments.

NEW QUESTION: 4

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. 5

D. 20

Answer: A (LEAVE A REPLY)

NEW QUESTION: 5

Which of the below datatype is not supported by Terraform.

A. Array

B. Map

C. List

D. Object

Answer: (SHOW ANSWER)

NEW QUESTION: 6

You want terraform plan and apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?

A. Local Backends

B. This can be done using any of the local or remote backends

C. Remote Backends

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

NEW QUESTION: 7
True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.
A. False
B. True
Explanation
Currently, Terraform has no mechanism to redact or protect secrets that are returned via data sources, so secrets read via this provider will be persisted into the Terraform state, into any plan files, and in some cases in the console output produced while planning and applying. These artifacts must, therefore, all be protected accordingly.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 8
How is terraform import run?
A. By an explicit call
B. As a part of terraform plan
C. All of the above
D. As a part of terraform refresh
E. As a part of terraform init
Answer: (SHOW ANSWER)

NEW QUESTION: 9
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: 10
HashiCorp Configuration Language (HCL) supports user-defined functions.
A. True
B. False
Answer: B (LEAVE A REPLY)

NEW QUESTION: 11
terraform refresh command will not modify infrastructure, but does modify the state file.
A. True
B. False
Answer: A (LEAVE A REPLY)
Explanation
The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. This can be used to detect any drift from the last-known state, and to update the state file. This does not modify infrastructure, but does modify the state file.
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION: 12
When does terraform apply reflect changes in the cloud environment?
A. However long it takes the resource provider to fulfill the request
B. Based on the value provided to the -refresh command line argument
C. Immediately
D. None of the above
E. After updating the state file
Answer: (SHOW ANSWER)

NEW QUESTION: 13
Terraform-specific settings and behaviors are declared in which configuration block type?
A. provider
B. terraform
C. resource
D. data

Answer: B (LEAVE A REPLY)
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: 14
What allows you to conveniently switch between multiple instances of a single configuration within its single backend?
A. Local backends
B. Providers
C. Remote backends
D. Workspaces
Answer: D (LEAVE A REPLY)
Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure.
Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION: 15
Where in your Terraform configuration do you specify a state backend?
A. The terraform block
B. The resource block
C. The provider block
D. The datasource block
Answer: A (LEAVE A REPLY)
Backends are configured with a nested backend block within the top-level terraform block.

NEW QUESTION: 16
You cannot publish your own modules on the Terraform Registry.
A. False
B. True
Answer: A (LEAVE A REPLY)

Explanation
Anyone can publish and share modules on the Terraform Registry.
https://www.terraform.io/docs/registry/modules/publish.html

NEW QUESTION: 17
What allows you to conveniently switch between multiple instances of a single configuration within its single backend?
A. Local backends
B. Providers
C. Remote backends
D. Workspaces
Answer: D (LEAVE A REPLY)
Explanation
Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure.
Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION: 18
During a terraform plan, a resource is successfully created but eventually fails during provisioning. What happens to the resource?
A. Terraform attempts to provision the resource up to three times before exiting with an error
B. the terraform plan is rolled back and all provisioned resources are removed
C. it is automatically deleted
D. the resource is marked as tainted
Answer: D (LEAVE A REPLY)
Explanation

If a resource successfully creates but fails during provisioning, Terraform will error and mark the resource as "tainted". A resource that is tainted has been physically created, but can't be considered safe to use since provisioning failed. Terraform also does not automatically roll back and destroy the resource during the apply when the failure happens, because that would go against the execution plan: the execution plan would've said a resource will be created, but does not say it will ever be deleted.

NEW QUESTION: 19
resource "aws_s3_bucket" "example" { bucket = "my-test-s3-terraform-bucket" ...} resource "aws_iam_role"
"test_role" { name = "test_role" ...}
Due to the way that the application code is written , the s3 bucket must be created before the test role is created , otherwise there will be a problem. How can you ensure that?
A. This will already be taken care of by terraform native implicit dependency. Nothing else needs to be done from your end.
B. Add explicit dependency using depends_on . This will ensure the correct order of resource creation.
C. Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.
D. This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.
Answer: (SHOW ANSWER)
Explanation
Use the depends_on meta-argument to handle hidden resource dependencies that Terraform can't automatically infer.
Explicitly specifying a dependency is only necessary when a resource relies on some other resource's behavior but doesn't access any of that resource's data in its arguments.

NEW QUESTION: 20
If a module declares a variable with a default, that variable must also be defined within the module.
A. False
B. True
Answer: (SHOW ANSWER)

NEW QUESTION: 21
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: (SHOW ANSWER)
Explanation
When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.
Terraform Cloud can also be used with local operations, in which case only state is stored in the Terraform Cloud backend.
https://www.terraform.io/docs/backends/types/remote.html

NEW QUESTION: 22
Which of the following is not an action performed by terraform init?
A. Retrieve the source code for all referenced modules
B. Initialize a configured backend
C. Create a sample main.tf file
D. Load required provider plugins
Answer: (SHOW ANSWER)

NEW QUESTION: 23
True or False: Workspaces provide identical functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.
A. True
B. False
Explanation
https://www.terraform.io/docs/cloud/workspaces/index.html
https://www.terraform.io/docs/state/workspaces.html
Answer: B (LEAVE A REPLY)

NEW QUESTION: 24
What kind of resource dependency is stored in terraform.tfstate file?
A. Both implicit and explicit dependencies are stored in state file.
B. Only explicit dependencies are stored in state file.
C. Only implicit dependencies are stored in state file.
D. No dependency information is stored in state file.
Answer: A (LEAVE A REPLY)
Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a

resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state.
https://www.terraform.io/docs/state/purpose.html#metadata

NEW QUESTION: 25
Which of the below configuration file formats are supported by Terraform? (Select TWO)
A. Node
B. JSON
C. Go
D. YAML
E. HCL
Answer: (SHOW ANSWER)
Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations.
https://www.terraform.io/docs/configuration/

NEW QUESTION: 26
What are some of the features of Terraform state? (select three)
A. determining the correct order to destroy resources
B. inspection of cloud resources
C. mapping configuration to real-world resources
D. increased performance
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 27
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: 28
Which option can not be used to keep secrets out of Terraform configuration files?
A. secure string
B. A -var flag
C. Environment variables
D. A Terraform provider
Answer: (SHOW ANSWER)

NEW QUESTION: 29
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: 30
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: 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 plan
C. terraform output
D. terraform validate
E. terraform state
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 32
When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.
For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.
A. Export the value using terraform export and input the value using terraform input.
B. Configure the pertinent provider's configuration with a list of possible IP addresses to use.
C. Configure an output value in the application module in order to use that value for the DNS module.
D. Preconfigure the IP address as a parameter in the DNS module.
Answer: C (LEAVE A REPLY)
Output values are like the return values of a Terraform module, and have several uses:
* A child module can use outputs to expose a subset of its resource attributes to a parent module.
* A root module can use outputs to print certain values in the CLI output after running terraform apply.
* When using remote state, root module outputs can be accessed by other configurations via a terraform_remote_state data source.
https://www.terraform.io/docs/configuration/outputs.html

NEW QUESTION: 33
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: 34
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)
The environment variable must be in the format TF_VAR_name, so for the
https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION: 35
You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.
A. Run the command export TF_LOG_FILE=./terraform.log.
B. Run the command export TF_LOG_PATH=./terraform.log.
C. Run the command export TF_DEBUG_PATH=./terraform.log.
D. No explicit action required. Terraform will take care of this as you have enable TF_LOG.
Answer: (SHOW ANSWER)
https://www.terraform.io/docs/commands/environment-variables.html

NEW QUESTION: 36
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. Dynamic Blocks
C. Terraform Expression
D. Terraform Functions
Answer: (SHOW ANSWER)

NEW QUESTION: 37
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 = var(num_servers)
B. servers = num_servers
C. servers = var.num_servers

D. servers = variable.num_servers
Answer: B (LEAVE A REPLY)

NEW QUESTION: 38
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: D (LEAVE A REPLY)
Explanation
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: 39
Refer to the below code where developer is outputting the value of the database password but has used sensitive parameter to hide the output value in the CLI.
output "db_password" { value = aws_db_instance.db.password description = "The password for logging in to the database." sensitive = true} Since sensitive is set to true, the value associated with db password will not be present in state file as plain-text?
A. False
B. True
Answer: A (LEAVE A REPLY)
Explanation
Sensitive output values are still recorded in the state, and so will be visible to anyone who is able to access the state data.

NEW QUESTION: 40
Refer to the following terraform variable definition
variable "track_tag" { type = list default = ["data_ec2","integration_ec2","digital_ec2"]}
track_tag = { Name = element(var.track_tag,count.index)} If count.index is set to 2, which of the following values will be assigned to the name attribute of track_tag variable?

A. digital_ec2
B. data_ec2
C. track_tag
D. integration_ec2
Answer: (SHOW ANSWER)

NEW QUESTION: 41
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. private module registry
B. Terraform registry
C. custom VCS integration
D. remote runs
Answer: (SHOW ANSWER)

NEW QUESTION: 42
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. Terraform needs to install the necessary plugins first
D. The Terraform CLI needs you to log into Terraform cloud first
Answer: (SHOW ANSWER)

NEW QUESTION: 43
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)
Explanation
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.

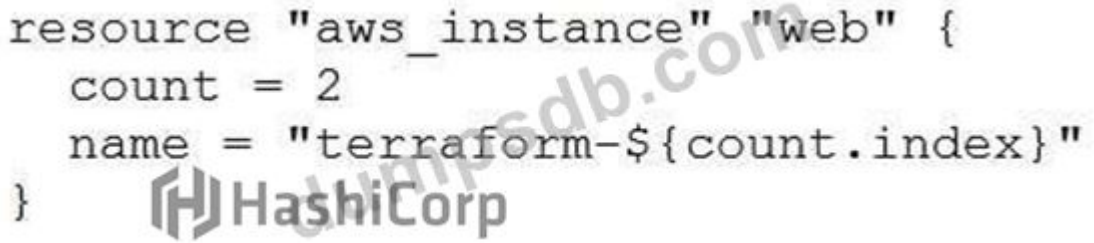
NEW QUESTION: 44
If a module uses a local variable, you can expose that value with a terraform output.
A. True
B. False

Answer: (SHOW ANSWER)
Explanation
Output values are like function return values.
Reference: https://www.terraform.io/docs/language/values/locals.html
https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION: 45
Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 ...) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region.
What should you do? Your provider setting is as below -
The default provider configuration provider "aws" { region = "us-east-1" }
A. No way to enable this via a single script . Write 2 different scripts with different default providers in the
2 scripts , one for us-east , another for us-west.
B. Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.
C. Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).
D. Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.
Answer: C (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"
}
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION: 46
How would you reference the "name" value of the second instance of this fictitious resource?



A. element(aws_instance.web, 2)
B. aws_instance.web[1].name
C. aws_instance.web[1]
D. aws_instance.web[2].name
E. aws_instance.web.*.name
Answer: A (LEAVE A REPLY)
Explanation/Reference: https://www.terraform.io/docs/configuration-0-11/interpolation.html

NEW QUESTION: 47
Select all Operating Systems that Terraform is available for. (select five)
A. Solaris
B. Unix
C. Windows
D. macOS
E. Linux
F. FreeBSD
Explanation
Terraform is available for macOS, FreeBSD, OpenBSD, Linux, Solaris, Windows
https://www.terraform.io/downloads.html
Answer: A,C,D,E,F (LEAVE A REPLY)

NEW QUESTION: 48
Using multi-cloud and provider-agnostic tools provides which of the following benefits?
A. Operations teams only need to learn and manage a single tool to manage infrastructure, regardless of where the infrastructure is deployed.
B. Increased risk due to all infrastructure relying on a single tool for management.
C. Can be used across major cloud providers and VM hypervisors.
D. Slower provisioning speed allows the operations team to catch mistakes before they are applied.
Answer: A,C (LEAVE A REPLY)
Using a tool like Terraform can be advantageous for organizations deploying workloads across multiple public and private cloud environments. Operations teams only need to learn a single tool, single language, and can use the same tooling to enable a DevOps-like experience and workflows.

NEW QUESTION: 49
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
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
Answer: (SHOW ANSWER)

NEW QUESTION: 50
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: (SHOW ANSWER)

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

NEW QUESTION: 51
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. test
D. azurerm
Answer: (SHOW ANSWER)

NEW QUESTION: 52
During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?
A. The resource will be planned for destruction and recreation upon the next terraform apply
B. Terraform will retry to provision again.
C. The failure of provisioner will be ignored and it will not cause a failure to terraform apply
D. The resource will be automatically destroyed.
Answer: A (LEAVE A REPLY)
If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply. Terraform does this because a failed provisioner can leave a resource in a semi-configured state. Because Terraform cannot reason about what the provisioner does, the only way to ensure proper creation of a resource is to recreate it. This is tainting.
You can change this behavior by setting the on_failure attribute, which is covered in detail below.
https://www.terraform.io/docs/provisioners/index.html#creation-time-provisioners
https://www.terraform.io/docs/provisioners/index.html#destroy-time-provisioners
https://www.terraform.io/docs/provisioners/index.html#failure-behavior

NEW QUESTION: 53
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)
Explanation/Reference: https://secrethub.io/blog/secret-management-for-terraform/

NEW QUESTION: 54
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: 55
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: 56
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: A,C,F (LEAVE A REPLY)
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: 57
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: (SHOW ANSWER)
Explanation
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION: 58
Multiple provider instances blocks for AWS can be part of 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: 59
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: 60
You have declared an input variable called environment in your parent module. What must you do to pass the value to a child module in the configuration?
A. Declare the variable in a terraform.tfvars file
B. Declare a node_count input variable for child module
C. Nothing, child modules inherit variables of parent module
D. Add node_count = var.node_count
Answer: B (LEAVE A REPLY)

NEW QUESTION: 61
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)
If you and your team are using Terraform to manage meaningful infrastructure, we recommend using the remote backend with Terraform Cloud or Terraform Enterprise.

NEW QUESTION: 62
Taint the resource "aws_instance" "baz" resource that lives in module bar which lives in module foo.
A. terraform taint module.foo.module.bar.baz
B. terraform taint module.foo.bar.aws_instance.baz
C. terraform taint module.foo.module.bar.aws_instance.baz
D. terraform taint foo.bar.aws_instance.baz
Answer: C (LEAVE A REPLY)
Explanation
Check resource addressing
https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION: 63
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: 64
You should store secret data in the same version control repository as your Terraform configuration.
A. False
B. True
Answer: A (LEAVE A REPLY)

NEW QUESTION: 65
What is the provider for this fictitious resource?

```
resource "aws_vpc" "main" {
    name = "test"
}
```

A. aws
B. vpc
C. main
D. test
Answer: A (LEAVE A REPLY)

NEW QUESTION: 66
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: 67
When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
A. False
B. True
Answer: B (LEAVE A REPLY)
TF_LOG_PATH specifies where the log should persist its output to. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
For example, to always write the log to the directory you're currently running terraform from:
export TF_LOG_PATH=./terraform.log
export TF_LOG=TRACE

NEW QUESTION: 68
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)
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: 69
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.
A. False
B. True
Answer: B (LEAVE A REPLY)

NEW QUESTION: 70
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 distinct list of route table name objects
C. Create a map of route table names from a list of subnet names
D. Create a map of route table names to subnet names
Answer: A (LEAVE A REPLY)

NEW QUESTION: 71

Which one of the following command will rewrite Terraform configuration files to a canonical format and style.

A. terraform graph -h

B. terraform init

C. terraform graph

D. terraform fmt

Answer: (SHOW ANSWER)

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

NEW QUESTION: 72

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. tainted and not destroyed

C. destroyed and recreated

D. recreated and tainted

Answer: (SHOW ANSWER)

NEW QUESTION: 73

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: 74
Which of the following value will be accepted for var1?
variable "var1" {
type = string
}
A. None of the above
B. Both A and B
C. "5"
D. 5
Answer: B (LEAVE A REPLY)
Explanation
Terraform automatically converts number and bool values to strings when needed.

NEW QUESTION: 75
You have created a custom variable definition file testing.tfvars. How will you use it for provisioning infrastructure?
A. terraform apply -var-state-file ="testing.tfvars"
B. terraform plan -var-file="testing.tfvar"
C. terraform apply -var-file="testing.tfvars"
D. terraform apply var-file="testing.tfvars"
Answer: C (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION: 76
You cannot publish your own modules on the Terraform Registry.
A. False
B. True
Answer: A (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/registry/modules/publish.html

NEW QUESTION: 77
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: (SHOW ANSWER)
S3 backend support state locking using DynamoDB.
https://www.terraform.io/docs/state/locking.html

NEW QUESTION: 78
Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)
A. CentOS
B. Red Hat
C. Amazon Linux
D. Unix
E. Ubuntu
Explanation
https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements
Answer: A,B,C,E (LEAVE A REPLY)

NEW QUESTION: 79
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)
Explanation
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 80
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: (SHOW ANSWER)
Explanation
The Terraform language will automatically convert number and bool values to string values when needed, and vice-versa as long as the string contains a valid representation of a number or boolean value.
Example
* true converts to "true", and vice-versa
* false converts to "false", and vice-versa
* 15 converts to "15", and vice-versa
Where possible, Terraform automatically converts values from one type to another in order to produce the expected type. If this isn't possible, Terraform will produce a type mismatch error and you must update the configuration with a more suitable expression.
https://www.terraform.io/docs/configuration/expressions.html#type-conversion

NEW QUESTION: 81
When using remote state, state is only ever held in memory when used by Terraform.
A. False
B. True
Answer: B (LEAVE A REPLY)

NEW QUESTION: 82
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: 83
Which of the following Terraform files should be ignored by Git when committing code to a repo? (select Three)
A. Files named exactly terraform.tfvars or terraform.tfvars.json.
B. Any files with names ending in .auto.tfvars or .auto.tfvars.json.
C. input.tf
D. terraform.tfstate
E. output.tf
Answer: A,B,D (LEAVE A REPLY)
Explanation
The .gitignore file should be configured to ignore Terraform files that either contain sensitive data or are not required to save.
Terraform state (terraform.tfstate) can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.
When using local state, state is stored in plain-text JSON files.
The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others.

NEW QUESTION: 84
You have created an AWS EC2 instance of type t2.micro through your terraform configuration file ec2.tf .
Now you want to change the instance type from t2.micro to t2.medium. Accordingly you have changed your configuration file and and ran terraform plan. After running terraform plan you check the output and saw one instance will be updated from t2.micro --> t2.medium. After this you went to grab a coffee without running terraform apply and meanwhile a member of your team changed the instance type of that EC2 instance to t2.medium from aws console. After coming to your desk you run terraform apply. What will happen?
A. The instance type will be changed to t2.micro and again will be changed to t2.medium
B. terraform apply will through an error.
C. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.
D. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
Answer: (SHOW ANSWER)

NEW QUESTION: 85

Which of the following command can be used to view the specified version constraints for all providers used in the current configuration.

A. terraform providers

B. terraform state show

C. terraform provider

D. terraform plan

Answer: (SHOW ANSWER)

Explanation

Use the terraform providers command to view the specified version constraints for all providers used in the current configuration.

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

NEW QUESTION: 86

What does the command terraform fmt do?

A. Rewrite Terraform configuration files to a canonical format and style.

B. Deletes the existing configuration file.

C. Updates the font of the configuration file to the official font supported by HashiCorp.

D. Formats the state file in order to ensure the latest state of resources can be obtained.

Answer: (SHOW ANSWER)

Explanation

The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style.

This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability.

Other Terraform commands that generate Terraform configuration will produce configuration files that conform to the style imposed by terraform fmt, so using this style in your own files will ensure consistency.

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

NEW QUESTION: 87

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: 88
Provisioners should only be used as a last resort.
A. False
B. True
Answer: B (LEAVE A REPLY)
Provisioners are a Last Resort
Terraform includes the concept of provisioners as a measure of pragmatism, knowing that there will always be certain behaviors that can't be directly represented in Terraform's declarative model.
However, they also add a considerable amount of complexity and uncertainty to Terraform usage. Firstly, Terraform cannot model the actions of provisioners as part of a plan because they can in principle take any action. Secondly, successful use of provisioners requires coordinating many more details than Terraform usage usually requires: direct network access to your servers, issuing Terraform credentials to log in, making sure that all of the necessary external software is installed, etc.
The following sections describe some situations which can be solved with provisioners in principle, but where better solutions are also available. We do not recommend using provisioners for any of the use-cases described in the following sections.
Even if your specific use-case is not described in the following sections, we still recommend attempting to solve it using other techniques first, and use provisioners only if there is no other option.
https://www.terraform.io/docs/provisioners/index.html

NEW QUESTION: 89
Select all features which are exclusive to Terraform Enterprise. (Select Three)
A. Sentinel
B. Cost Estimation
C. Audit Logs
D. Clustering
E. SAML/SSO
Answer: (SHOW ANSWER)
Sentinel and Cost Estimation are also available in Terraform Cloud
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION: 90
You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?
A. It depends on which command issue related to.
B. Python

C. Go
D. Java
Answer: C (LEAVE A REPLY)
Basic programming knowledge. Terraform and Terraform Plugins are written in the Go programming language, but even if you've never written a line of Go before, you're still welcome to take a dive into the code and submit patches. The community is happy to assist with code reviews and offer guidance specific to Go.

NEW QUESTION: 91
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: 92
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: 93
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: 94

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: A (LEAVE A REPLY)
Reference: https://github.com/hashicorp/terraform/issues/19291

NEW QUESTION: 95
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: (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.
https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION: 96
Which of the following allows Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code?
A. Sentinel
B. Module registry
C. Functions
D. Workspaces
Answer: A (LEAVE A REPLY)
Explanation
Sentinel is a language and framework for policy built to be embedded in existing software to enable fine-grained, logic-based policy decisions. A policy describes under what circumstances certain behaviors are allowed. Sentinel is an enterprise-only feature.
https://www.youtube.com/watch?v=Vy8s7AAvU6g&feature=emb_title

NEW QUESTION: 97
Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider.
Select all the supported VCS providers from the answers below. (select four)
A. GitHub

B. CVS Version Control
C. Azure DevOps Server
D. Bitbucket Cloud
E. GitHub Enterprise
Answer: A,C,D,E (LEAVE A REPLY)
Explanation
Terraform Cloud supports the following VCS providers:
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github-enterprise.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-com.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html
https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers

NEW QUESTION: 98
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: 99
terraform validate validates the syntax of Terraform files.
A. True
B. False
Answer: A (LEAVE A REPLY)
Explanation
The terraform validate command validates the syntax and arguments of the Terraform configuration files.
Reference: https://www.terraform.io/docs/cli/code/index.html

NEW QUESTION: 100
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)
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: 101
Which of the following is allowed as a Terraform variable name?
A. version
B. count
C. name
D. source
Answer: C (LEAVE A REPLY)

NEW QUESTION: 102
One remote backend configuration always maps to a single remote workspace.
A. False
B. True
Answer: (SHOW ANSWER)

NEW QUESTION: 103
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: 104
Why might a user opt to include the following snippet in their configuration file?
A. versions before Terraform 0.12 were not approved by HashiCorp to be used in production
B. this ensures that all Terraform providers are above a certain version to match the application being deployed
C. Terraform 0.12 introduced substantial changes to the syntax used to write Terraform configuration
D. The user wants to ensure that the application being deployed is a minimum version of 0.12
Answer: C (LEAVE A REPLY)

NEW QUESTION: 105
How does Terraform handle working with so many providers?
A. Terraform ships with all of the plugins embedded in the Terraform binary.
B. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in the configuration's working directory.
C. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in a shared, system-wide plugins directory.
D. Terraform allows you to select the providers you want to support during the Terraform installation process.
Answer: (SHOW ANSWER)
Explanation
Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins, even the core types such as AWS and Heroku. Users of Terraform are able to write new plugins in order to support new functionality in Terraform.

NEW QUESTION: 106

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)

NEW QUESTION: 107
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: 108
What is the name assigned by Terraform to reference this resource?

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

A. test
B. dev
C. azurerm_resource_group
D. azurerm
Answer: (SHOW ANSWER)

NEW QUESTION: 109
After creating a new workspace "PROD" you need to run the command terraform select PROD to switch to it.

A. False
B. True
Answer: A (LEAVE A REPLY)
By default, when you create a new workspace you are automatically switched to it To create a new workspace and switch to it, you can use terraform workspace new <new_workspace_name>; to switch to a existing workspace you can use terraform workspace select <existing_workspace_name>; Example:
$ terraform workspace new example
Created and switched to workspace "example"!
You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.

NEW QUESTION: 110
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: (SHOW ANSWER)
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: 111
Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 ...) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region. What should you do? Your provider setting is as below -
The default provider configuration provider "aws" { region = "us-east-1" }
A. No way to enable this via a single script . Write 2 different scripts with different default providers in the 2 scripts , one for us-east , another for us-west.
B. Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.

C. Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).
D. Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.
Answer: C (LEAVE A REPLY)
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: 112
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. non-traditional
Explanation
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.
B. disconnected
C. a private install
D. air-gapped
Answer: A (LEAVE A REPLY)

NEW QUESTION: 113

A Terraform local value can reference other Terraform local values.
A. True
B. False
Answer: A (LEAVE A REPLY)

NEW QUESTION: 114
Terraform providers are always installed from the Internet.
A. True
B. False
Answer: (SHOW ANSWER)
Terraform configurations must declare which providers they require, so that Terraform can install and use them.

NEW QUESTION: 115
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)
Explanation
https://www.terraform.io/docs/provisioners/index.html

NEW QUESTION: 116
Terraform variables and outputs that set the "description" argument will store that description in the state file.
A. True
B. False
Answer: (SHOW ANSWER)
Reference: https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION: 117
Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)
A. Explicit use of a provider block in configuration, optionally including a version constraint.
B. Use of any resource belonging to a particular provider in a resource or data block in configuration.
C. Existence of any resource instance belonging to a particular provider in the current state.
D. Existence of any provider plugins found locally in the working directory.

Answer: A,B,C (LEAVE A REPLY)
Explanation
The existence of a provider plugin found locally in the working directory does not itself create a provider dependency. The plugin can exist without any reference to it in the terraform configuration. https://www.terraform.io/docs/commands/providers.html

NEW QUESTION: 118
What are the benefits of using Infrastructure as Code? (select five)
A. Infrastructure as Code easily replaces development languages such as Go and .Net for application development
B. Infrastructure as Code gives the user the ability to recreate an application's infrastructure for disaster recovery scenarios
C. Infrastructure as Code is relatively simple to learn and write, regardless of a user's prior experience with developing code
D. Infrastructure as Code provides configuration consistency and standardization among deployments
E. Infrastructure as Code is easily repeatable, allowing the user to reuse code to deploy similar, yet different resources
F. 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.
Answer: B,C,E,F (LEAVE A REPLY)

NEW QUESTION: 119
In the example below, the depends_on argument creates what type of dependency?
A. explicit dependency
B. internal dependency
C. implicit dependency
D. non-dependency resource
Answer: D (LEAVE A REPLY)

NEW QUESTION: 120
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: 121
Which of the following commands will launch the Interactive console for Terraform interpolations?
A. terraform
B. terraform console
C. terraform cmdline
Explanation
https://www.terraform.io/docs/commands/console.html
D. terraform cli
Answer: D (LEAVE A REPLY)

NEW QUESTION: 122
Which of the following is an invalid variable name?
A. count
B. instance_name
Explanation
https://www.terraform.io/intro/examples/count.html
C. web
D. var1
Answer: A (LEAVE A REPLY)

NEW QUESTION: 123
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: 124
Your manager has instructed you to start using terraform for your day-to-day operations, but your security team is concerned about the terraform state files. They have heard it contains confidential information, and are worried that it will not be securely protected. What should be your response to the security team in this regard?
A. Inform the security team that using terraform state is optional . There are ways to avoid it , and you will do the same.
B. Ensure that the state is managed in a remote backend , preferably an enterprise grade state management system like Terraform Cloud.
C. Mask the confidential entries in the terraform state file , using Vault Enterprise, another Hashicorp product , while keeping it locally.
D. Keep the state file locally on each developer machine , and ensure that there is a local protection software like KeyPass protecting it.
Answer: B (LEAVE A REPLY)
https://www.terraform.io/docs/state/index.html
State is very important topic for exam. Please read all of the below subtopics Purpose Import Existing Resources Locking Workspaces Remote State Sensitive Data

NEW QUESTION: 125
How can you trigger a run in a Terraform Cloud workspace that is connected to a Version Control System (VCS) repository?
A. Commit a change to the VCS working directory and branch that the Terraform Cloud workspace is connected to
B. Only Terraform Cloud organization owners can set workspace variables on VCS connected workspaces
C. Only Terraform Cloud organization owners can approve plans in VCS connected workspaces
D. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces
Answer: A (LEAVE A REPLY)

NEW QUESTION: 126
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: 127
Matt wants to import a manually created EC2 instance into terraform so that he can manage the EC2 instance through terraform going forward. He has written the configuration file of the EC2 instance before importing it to Terraform. Following is the code:
resource "aws_instance" "matt_ec2" { ami = "ami-bg2640de" instance_type = "t2.micro" vpc_security_group_ids = ["sg-6ae7d613", "sg-53370035"] key_name = "mysecret" subnet_id =
"subnet-9e3cfbc5" }
The instance id of that EC2 instance is i-0260835eb7e9bd40 How he can import data of EC2 to state file?
A. terraform import aws_instance.id = i-0260835eb7e9bd40
B. terraform import i-0260835eb7e9bd40
C. terraform import aws_instance.i-0260835eb7e9bd40
D. terraform import aws_instance.matt_ec2 i-0260835eb7e9bd40
Answer: D (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/import/usage.html

NEW QUESTION: 128
If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?
A. Run terraform refresh
B. Run terraform import
C. It will happen automatically
D. Manually update the state fire
Answer: C (LEAVE A REPLY)

NEW QUESTION: 129
A terraform apply can not _____ infrastructure.
A. provision
B. change
C. import
D. destroy

Answer: (SHOW ANSWER)

NEW QUESTION: 130
You have declared a variable called var.list which is a list of objects that all have an attribute id.
Which options will produce a list of the IDs? (Choose two.)
A. var.list[*].id
B. [for o in var.list : o.id]
C. [var.list[*].id]
D. { for o in var.list : o => o.id }
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 131
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: 132
The terraform.tfstate file always matches your currently built infrastructure.
A. True
B. False
Answer: B (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/language/state/index.html

NEW QUESTION: 133
What is the workflow for deploying new infrastructure with Terraform?
A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.
C. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.
Answer: C (LEAVE A REPLY)

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

NEW QUESTION: 134
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: 135

A user has created three workspaces using the command line - prod, dev, and test. The user wants to create a fourth workspace named stage. Which command will the user execute to accomplish this?
A. terraform workspace new stage
B. terraform workspace -new stage
C. terraform workspace -create stage
D. terraform workspace create stage
Answer: (SHOW ANSWER)
The terraform workspace new command is used to create a new workspace.
https://www.terraform.io/docs/commands/workspace/new.html

NEW QUESTION: 136
Which flag would you add to terraform plan to save the execution plan to a file?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.
Answer:
outFILENAME

NEW QUESTION: 137
A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip"} What is the best way to ensure that you are not using the latest version of the module?
A. Add a module version constraint in your configuration's backend block and specify a previous version.
B. Add a version key to the module configuration and specify a previous version.
C. Delete the latest version of the module in S3 to rollback to the previous version.

D. Add a version property to the module in Terraform's state file and specify a previous version.

Answer: C (LEAVE A REPLY)

Only Terraform Registries support module versioning by using the version key, one cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version.

https://www.terraform.io/docs/configuration/modules.html#source

NEW QUESTION: 138

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: A (LEAVE A REPLY)

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: 139

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

A. Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using Policy-As-Code.

B. Implement a review process where every code will be reviewed before merging to the master branch.

C. Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and write Policy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.

D. Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.

Answer: C (LEAVE A REPLY)

Explanation

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

NEW QUESTION: 140

If a module uses a local variable, you can expose that value with a terraform output.

A. True

B. False

Answer: A (LEAVE A REPLY)

Output values are like function return values.

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

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

NEW QUESTION: 141

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: 142
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: 143
While Terraform is generally written using the HashiCorp Configuration Language (HCL), what other syntax can Terraform are expressed in?
A. JSON
B. YAML
C. TypeScript
D. XML
Answer: A (LEAVE A REPLY)
Explanation
The constructs in the Terraform language can also be expressed in JSON syntax, which is harder for humans to read and edit but easier to generate and parse programmatically.

NEW QUESTION: 144
You are reviewing Terraform configurations for a big project in your company. You noticed that there are several identical sets of resources that appear in multiple configurations. What feature of Terraform would you recommend to use to reduce the amount of cloned configuration between the different configurations?
A. Packages
B. Backends
C. Provisioners
D. Modules
Answer: D (LEAVE A REPLY)

Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.
A module is a container for multiple resources that are used together. Modules can be used to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.
Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.
https://www.terraform.io/docs/modules/index.html

NEW QUESTION: 145
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)
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: 146
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. Which of the below option is a recommended solution for this?
A. Remote State
B. Module
C. Use the cached state and treat this as the record of truth.
D. Workspace
Answer: A (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/state/remote.html

NEW QUESTION: 147
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.
Reference: https://www.terraform.io/docs/cli/commands/force-unlock.html

NEW QUESTION: 148
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: 149
A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip"} What is the best way to ensure that you are not using the latest version of the module?
A. Add a module version constraint in your configuration's backend block and specify a previous version.
B. Add a version key to the module configuration and specify a previous version.
C. Delete the latest version of the module in S3 to rollback to the previous version.
D. Add a version property to the module in Terraform's state file and specify a previous version.
Answer: C (LEAVE A REPLY)
Explanation
Version constraints are supported only for modules installed from a module registry, such as the Terraform Registry or Terraform Cloud's private module registry. Other module sources can provide their own versioning mechanisms within the source string itself, or might not support versions at all. In particular, modules sourced from local file paths do not support version; since they're loaded from the same source repository.
Only Terraform Registries support module versioning by using the version key, one cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use

the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version.
https://www.terraform.io/docs/configuration/modules.html#source

NEW QUESTION: 150
Terraform provisioners can be added to any resource block.
A. False
B. True
Answer: (SHOW ANSWER)

NEW QUESTION: 151
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: 152
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: (SHOW ANSWER)
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: 153
What is the default backend for Terraform?
A. consul
B. gcs
C. local
D. etcd
Answer: C (LEAVE A REPLY)
Explanation
By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to.
https://www.terraform.io/docs/backends/index.html

NEW QUESTION: 154
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: 155
You have declared a variable name my_var in terraform configuration without a value associated with it.
variable my_var {}
After running terraform plan it will show an error as variable is not defined.
A. True
B. False
Answer: B (LEAVE A REPLY)
Explanation
Input variables are usually defined by stating a name, type and a default value. However, the type and default values are not strictly necessary. Terraform can deduct the type of the variable from the default or input value.
Variables can be predetermined in a file or included in the command-line options. As such, the simplest variable is just a name while the type and value are selected based on the input.

```
variable "variable_name" {}
terraform apply -var variable_name="value"
```

The input variables, like the one above, use a couple of different types: strings, lists, maps, and boolean. Here are some examples of how each type are defined and used.

String

Strings mark a single value per structure and are commonly used to simplify and make complicated values more user-friendly. Below is an example of a string variable definition.

```
variable "template" {
type = string
default = "01000000-0000-4000-8000-000030080200"
}
```

A string variable can then be used in resource plans. Surrounded by double quotes, string variables are a simple substitution such as the example underneath.

```
storage = var.template
```

List

Another type of Terraform variables lists. They work much like a numbered catalogue of values. Each value can be called by their corresponding index in the list. Here is an example of a list variable definition.

```
variable "users" {
type = list
default = ["root", "user1", "user2"]
}
```

Lists can be used in the resource plans similarly to strings, but you'll also need to denote the index of the value you are looking for.

```
username = var.users[0]
```

Map

Maps are a collection of string keys and string values. These can be useful for selecting values based on predefined parameters such as the server configuration by the monthly price.

```
variable "plans" {
type = map
default = {
"5USD" = "1xCPU-1GB"
"10USD" = "1xCPU-2GB"
"20USD" = "2xCPU-4GB"
}
}
```

You can access the right value by using the matching key. For example, the variable below would set the plan to "1xCPU-1GB".

```
plan = var.plans["5USD"]
```

The values matching to their keys can also be used to look up information in other maps. For example, underneath is a shortlist of plans and their corresponding storage sizes.

```
variable "storage_sizes" {
type = map
default = {
"1xCPU-1GB" = "25"
"1xCPU-2GB" = "50"
"2xCPU-4GB" = "80"
}
}
```

These can then be used to find the right storage size based on the monthly price as defined in the previous example.

```
size = lookup(var.storage_sizes, var.plans["5USD"])
```

Boolean

The last of the available variable type is boolean. They give the option to employ simple true or false values.

For example, you might wish to have a variable that decides when to generate the root user password on a new deployment.

```
variable "set_password" {
default = false
}
```

The above example boolean can be used similarly to a string variable by simply marking down the correct variable.

```
create_password = var.set_password
```

By default, the value is set to false in this example. However, you can overwrite the variable at deployment by assigning a different value in a command-line variable.

```
terraform apply -var set_password="true"
```

NEW QUESTION: 156

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: 157

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)
Explanation
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: 158
When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.
For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.
A. Export the value using terraform export and input the value using terraform input.
B. Configure the pertinent provider's configuration with a list of possible IP addresses to use.
C. Configure an output value in the application module in order to use that value for the DNS module.
D. Preconfigure the IP address as a parameter in the DNS module.
Answer: C (LEAVE A REPLY)
Explanation
Output values are like the return values of a Terraform module, and have several uses:
* A child module can use outputs to expose a subset of its resource attributes to a parent module.
* A root module can use outputs to print certain values in the CLI output after running terraform apply.
* When using remote state, root module outputs can be accessed by other configurations via a terraform_remote_state data source.
https://www.terraform.io/docs/configuration/outputs.html

NEW QUESTION: 159
Every region in AWS has a different AMI ID for Linux and these are keep on changing.
What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?
A. Use data source aws_ami.
B. Create a map of region to ami id.

C. Create different configuration file for different region.
D. None of the above
Answer: A (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/configuration/data-sources.html

NEW QUESTION: 160
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. CLI access from the same machine running Terraform
C. integration with a tool like Jenkins
D. Vault provider
Answer: D (LEAVE A REPLY)

NEW QUESTION: 161
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 question TF_VAR_var1 is the correct choice.
https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION: 162
Terraform works well in Windows but a Windows server is required.
A. True
B. False
Answer: B (LEAVE A REPLY)

NEW QUESTION: 163
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: 164
When using providers that require the retrieval of data, such as the HashiCorp Vault provider, in what phase does Terraform actually retrieve the data required?
A. terraform init
B. terraform apply
C. terraform plan
D. terraform delete
Answer: A (LEAVE A REPLY)

NEW QUESTION: 165
What is the result of the following terraform function call?
A. hello
B. what?
C. goodbye
Answer: (SHOW ANSWER)
Explanation
https://www.terraform.io/docs/configuration/functions/lookup.html

NEW QUESTION: 166
Which of the following Terraform files should be ignored by Git when committing code to a repo? (select Three)
A. Files named exactly terraform.tfvars or terraform.tfvars.json.
B. Any files with names ending in .auto.tfvars or .auto.tfvars.json.
C. input.tf
D. terraform.tfstate
E. output.tf
Answer: A,B,D (LEAVE A REPLY)
The .gitignore file should be configured to ignore Terraform files that either contain sensitive data or are not required to save.
Terraform state (terraform.tfstate) can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.
When using local state, state is stored in plain-text JSON files.
The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others.

NEW QUESTION: 167
terraform init initializes a sample main.tf file in the current directory.
A. True
B. False
Answer: B (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/cli/commands/init.html

NEW QUESTION: 168
A data block requests that Terraform read from a given data source and export the result under the given local name.
A. True
B. False
Answer: A (LEAVE A REPLY)

NEW QUESTION: 169
A Terraform provisioner must be nested inside a resource configuration block.
A. True
B. False
Answer: A (LEAVE A REPLY)
Explanation
Most provisioners require access to the remote resource via SSH or WinRM, and expect a nested connection block with details about how to connect.
Reference: https://www.terraform.io/docs/language/resources/provisioners/connection.html

NEW QUESTION: 170
A provider configuration block is required in every Terraform configuration.
Example:

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

A. True
B. False

Answer: (SHOW ANSWER)
Reference: https://github.com/hashicorp/terraform/issues/17928

NEW QUESTION: 171
In regards to Terraform state file, select all the statements below which are correct?
A. When using local state, the state file is stored in plain-text.
B. The state file is always encrypted at rest.
C. Storing state remotely can provide better security.
D. Using the mask feature, you can instruct Terraform to mask sensitive data in the state file.
E. The Terraform state can contain sensitive data, therefore the state file should be protected from unauthorized access.
F. Terraform Cloud always encrypts state at rest.
Answer: A,C,E,F (LEAVE A REPLY)
Explanation
Terraform state can contain sensitive data, depending on the resources in use and your definition of
"sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.
When using local state, state is stored in plain-text JSON files.
When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.
Storing Terraform state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.
Recommendations
If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.
Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.
For example:
* Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.
* The S3 backend supports encryption at rest when the encrypt option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

NEW QUESTION: 172

You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.

A. Run the command export TF_LOG_FILE=./terraform.log.

B. Run the command export TF_LOG_PATH=./terraform.log.

C. Run the command export TF_DEBUG_PATH=./terraform.log.

D. No explicit action required. Terraform will take care of this as you have enable TF_LOG.

Answer: B (LEAVE A REPLY)

Explanation

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

NEW QUESTION: 173

1. resource "aws_s3_bucket" "example" {
2. bucket = "my-test-s3-terraform-bucket"
3. ...} resource "aws_iam_role" "test_role" {
4. name = "test_role"
5. ...}

Due to the way that the application code is written, the s3 bucket must be created before the test role is created, otherwise there will be a problem. How can you ensure that?

A. Add explicit dependency using depends_on . This will ensure the correct order of resource creation.

B. This will already be taken care of by terraform native implicit dependency. Nothing else needs to be done from your end.

C. This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.

D. Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.

Answer: A (LEAVE A REPLY)

Explanation

Implicit dependency works only if there is some reference of one resource to another. Explicit dependency is the option here.

NEW QUESTION: 174

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: 175

Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)
A. Validate
B. Apply
C. Destroy
D. Import
E. Init
Answer: B,E (LEAVE A REPLY)

NEW QUESTION: 176
Which of the following is not a valid Terraform collection type?
A. map
B. set
C. list
D. tree
Answer: D (LEAVE A REPLY)

NEW QUESTION: 177
Your manager has instructed you to start using terraform for the entire infra provisioning of the application stack. There are 4 environments - DEV , QA , UAT , and PROD. The application team has asked for complete segregation between these environments including the backend , state , and also configurations ,since there will be unique resources in different environments . What is the possible way to structure the terraform code to facilitate that.
A. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to a different backend.
B. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to the same backend.
C. Implement terraform workspaces , and map each environment with one workspace.
D. Enable remote backend storage . Configure 4 different backend storages , one for each environment.
Answer: A (LEAVE A REPLY)
In particular, organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g. staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are not a suitable isolation mechanism for this scenario.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION: 178
When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
A. False
B. True
Answer: (SHOW ANSWER)
Explanation
TF_LOG_PATH specifies where the log should persist its output to. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
For example, to always write the log to the directory you're currently running terraform from:
export TF_LOG_PATH=./terraform.log
export TF_LOG=TRACE

NEW QUESTION: 179
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?ref=v1.0.4"
Explanation
https://www.terraform.io/docs/modules/sources.html#selecting-a-revision
D. source = "git::https://example.com/my_test_module.git@tag=v1.0.4"
Answer: C (LEAVE A REPLY)

NEW QUESTION: 180
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: (SHOW ANSWER)
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: 181
Your company has a lot of workloads in AWS , and Azure that were respectively created using CloudFormation , and AzureRM Templates. However , now your CIO has decided to use Terraform for all new projects , and has asked you to check how to integrate the existing environment with terraform code. What should be your next plan of action?
A. Just write the terraform config file for the new resources , and run terraform apply , the state file will automatically be updated with the details of the new resources to be imported.
B. Use terraform import command to import each resource one by one .
C. Tell the CIO that this is not possible . Resources created in CloudFormation , and AzureRM templates cannot be tracked using terraform.
D. This is only possible in Terraform Enterprise , which has the TerraformConverter exe that can take any other template language like AzureRM and convert to Terraform code.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 182
One remote backend configuration always maps to a single remote workspace.
A. True
B. False
Answer: (SHOW ANSWER)
Explanation/Reference:
https://www.terraform.io/docs/language/settings/backends/remote.html

NEW QUESTION: 183
Terraform requires the Go runtime as a prerequisite for installation.
A. False
B. True
Answer: A (LEAVE A REPLY)

NEW QUESTION: 184
Why would you use the terraform taint command?
A. When you want to force Terraform to destroy a resource on the next apply
B. When you want to force Terraform to destroy and recreate a resource on the next apply
C. When you want Terraform to ignore a resource on the next apply
D. When you want Terraform to destroy all the infrastructure in your workspace
Answer: B (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.