HashiCorp.TA-002-P.v2021-04-20.q98

Exam Code:	TA-002-P
Exam Name:	HashiCorp Certified: Terraform Associate
Certification Provider:	HashiCorp
Free Question Number:	98
Version:	v2021-04-20
# of views:	2139
# of Questions views:	105115
https://www.freecram.com/torrent/HashiCorp.TA-002-P.v2021-04-20.q98.html	

NEW QUESTION: 1
Complete the following sentence:
For local state, the workspaces are stored directly in a _____.
A. a file called terraform.tfstate.backup
B. directory called terraform.workspaces.tfstate
C. a file called terraform.tfstate
D. directory called terraform.tfstate.d
Answer: (SHOW ANSWER)
Explanation
For local state, Terraform stores the workspace states in a directory called terraform.tfstate.d.
https://www.terraform.io/docs/state/workspaces.html#workspace-internals

NEW QUESTION: 2
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: A (LEAVE A REPLY)
Explanation
The terraform workspace new command is used to create a new workspace.
https://www.terraform.io/docs/commands/workspace/new.html

NEW QUESTION: 3

The Terraform language does not support user-defined functions, and so only the functions built in to the language are available for use.

A. False

B. True

Answer: B (LEAVE A REPLY)

Explanation

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

NEW QUESTION: 4

When does terraform apply reflect changes in the cloud environment?

A. None of the above

B. After updating the state file

C. However long it takes the resource provider to fulfill the request

D. Immediately

E. Based on the value provided to the -refresh command line argument

Answer: A (LEAVE A REPLY)

NEW QUESTION: 5

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: 6

Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.

A. Set TF_LOG = DEBUG

B. Set VAR_TF = TRACE

C. Set TF_LOG = TRACE

D. Set VAR_TF_LOG = TRACE

Answer: (SHOW ANSWER)

Explanation

https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 7

What Terraform feature is shown in the example below?

A. conditional expression

B. local values

C. dynamic block
D. data source
Answer: (SHOW ANSWER)

NEW QUESTION: 8
Your team lead does not trust the junior terraform engineers who now have access to the git repo . So , he wants you to have some sort of a checking layer , whereby , you can ensure that the juniors will not create any non-compliant resources that might lead to a security audit failure in future. What can you do to efficiently enforce this?
A. 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.
B. 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.
C. 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.
D. Create a git master branch , and implement PR . Every change needs to be reviewed by you , before being merged to the master branch.
Answer: B (LEAVE A REPLY)
Explanation
Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION: 9
Command terraform refresh will update state file?
A. False
B. True
Answer: B (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. If the state is changed, this may cause changes to occur during the next plan or apply.
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION: 10

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: 11

Terraform works well in Windows but a Windows server is required.

A. False

B. True

Answer: A (LEAVE A REPLY)

Explanation

You may see this question in actual exam. Please remember : Terraform does not require GO language to be installed as a prerequisite and it does not require a Windows Server as well.

NEW QUESTION: 12

terraform state subcommands such as list are read-only commands, do read-only commands create state backup files?

A. Yes

B. No

Answer: (SHOW ANSWER)

Explanation

Subcommands that are read-only (such as list) do not write any backup files since they aren't modifying the state.

All terraform state subcommands that modify the state write backup files. The path of these backup file can be controlled with -backup.

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

NEW QUESTION: 13
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: (SHOW ANSWER)
Explanation
Backends are configured with a nested backend block within the top-level terraform block.
Reference: https://www.terraform.io/docs/language/settings/backends/configuration.html

NEW QUESTION: 14
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: 15
By default, where does Terraform store its state file?
A. Amazon S3 bucket
B. shared directory
C. remotely using Terraform Cloud
D. current working directory
Answer: (SHOW ANSWER)
Explanation
By default, the state file is stored in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

NEW QUESTION: 16
A single terraform resource file that defines an aws_instance resource can simply be renamed to vsphere_virtual_machine in order to switch cloud providers.
A. True
B. False
Answer: B (LEAVE A REPLY)
Explanation
Every provider has its own required and allowed declarations none of which match between cloud providers.

NEW QUESTION: 17
Which of the following represents a feature of Terraform Cloud that is NOT free to customers?
A. Roles and Team Management
B. WorkSpace Management
C. Private Module Registry
D. VCS Integration
Answer: A (LEAVE A REPLY)
Explanation
Role Based Access Controls (RBAC) for controlling permissions for who has access to what configurations within an organization and it is not free to customers.
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION: 18
Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.
What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?
A. Submit a ticket to AWS and ask them to export the state of all existing resources and use terraform import to import them into the state file.
B. Delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward.
C. Resources that are manually deployed in the AWS console cannot be imported by Terraform.
D. Using terraform import, import the existing infrastructure into your Terraform state.
Answer: (SHOW ANSWER)
Explanation
Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management. This is a great way to slowly transition infrastructure to Terraform.
The terraform import command is used to import existing infrastructure.

To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.
Example:
resource "aws_instance" "import_example" {
# ...instance configuration...
}
Now terraform import can be run to attach an existing instance to this resource configuration.
$ terraform import aws_instance.import_example i-03efafa258104165f
aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f)
aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION: 19
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: 20

You run a local-exec provisioner in a null resource called null_resource.run_script and realize that you need to rerun the script.
Which of the following commands would you use first?
A. terraform apply -target=null_resource.run_script
B. terraform plan -target=null_resource.run_script
C. terraform validate null_resource.run_script
D. terraform taint null_resource.run_script
Answer: D (LEAVE A REPLY)

NEW QUESTION: 21
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: 22
You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season. Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.
How would you solve this using infrastructure as code?
A. Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource
B. Implement a checklist that engineers can follow when configuring compute instances
C. Replace the compute instance type with a larger version to reduce the number of required deployments

D. Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews
Answer: A (LEAVE A REPLY)

NEW QUESTION: 23
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 = variable.num_servers
C. servers = num_servers
D. servers = var(num_servers)
Answer: C (LEAVE A REPLY)

NEW QUESTION: 24
In Terraform Enterprise, a workspace can be mapped to how many VCS repos?
A. 5
B. 2
C. 3
D. 1
Answer: D (LEAVE A REPLY)
Explanation
A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.
https://www.terraform.io/docs/cloud/workspaces/vcs.html

NEW QUESTION: 25
State is a requirement for Terraform to function
A. True
B. False
Answer: (SHOW ANSWER)
Explanation
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run.
Purpose of Terraform State
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.
As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).

1. Mapping to the Real World

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

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

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

2. Metadata

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

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

To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state.

Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.

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

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

3. Performance

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

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

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

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
4. Syncing
In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but when using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects.
Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION: 26
Valarie has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code. Valarie wants to hide the output value in the CLI after terraform apply that's why she has used sensitive parameter.
1. output "db_password" {
2. value = local.db_password
3. sensitive = true
4. }
Since sensitive is set to true, will the value associated with db password be available in plain-text in the state file for everyone to read?
A. Yes
B. No
Answer: A (LEAVE A REPLY)
Explanation
Outputs can be marked as containing sensitive material by setting the sensitive attribute to true, like this:
output "sensitive" {
sensitive = true
value = VALUE
}
When outputs are displayed on-screen following a terraform apply or terraform refresh, sensitive outputs are redacted, with <sensitive> displayed in place of their value.
Limitations of Sensitive Outputs

The values of sensitive outputs are still stored in the Terraform state, and available using the terraform output command, so cannot be relied on as a sole means of protecting values.
Sensitivity is not tracked internally, so if the output is interpolated in another module into a resource, the value will be displayed.

NEW QUESTION: 27
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)
Explanation
Each time a new provider is added to configuration -- either explicitly via a provider block or by adding a resource from that provider -- Terraform must initialize the provider before it can be used. Initialization downloads and installs the provider's plugin so that it can later be executed.

NEW QUESTION: 28
What information does the public Terraform Module Registry automatically expose about published modules?
A. Required input variables
B. Optional inputs variables and default values
C. Outputs
D. All of the above
E. None of the above
Answer: E (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/registry/modules/publish.html

NEW QUESTION: 29
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)

Explanation
Github is not a supported backend type.
https://www.terraform.io/docs/backends/types/index.html

NEW QUESTION: 30
What is a downside to using the Vault provider to read secrets from Vault?
A. Secrets are persisted to the state file and plans.
B. Terraform and Vault must be running on the same version.
C. Terraform and Vault must be running on the same physical host.
D. Terraform requires a unique auth method to work with Vault.
Answer: A (LEAVE A REPLY)
Explanation
The Vault provider allows Terraform to read from, write to, and configure Hashicorp Vault. Interacting with Vault from Terraform causes any secrets that you read and write to be persisted in both Terraform's state file and in any generated plan files. For any Terraform module that reads or writes Vault secrets, these files should be treated as sensitive and protected accordingly.

NEW QUESTION: 31
What is the provider for this fictitious resource?

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


A. vpc
B. main
C. aws
D. test
Answer: C (LEAVE A REPLY)
Reference: https://docs.aws.amazon.com/cloudformation-cli/latest/userguide/resource-types.html

NEW QUESTION: 32
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)
Explanation
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: 33
Terraform variables and outputs that set the "description" argument will store that description in the state file.
A. True
B. False
Answer: B (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION: 34
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: 35
What are some of the problems of how infrastructure was traditionally managed before Infrastructure as Code?
(select three)
A. Requests for infrastructure or hardware required a ticket, increasing the time required to deploy applications
B. Traditional deployment methods are not able to meet the demands of the modern business where resources tend to live days to weeks, rather than months to years
C. Traditionally managed infrastructure can't keep up with cyclic or elastic applications
D. Pointing and clicking in a management console is a scalable approach and reduces human error as businesses are moving to a multi-cloud deployment model
Answer: A,B,C (LEAVE A REPLY)
Explanation
Businesses are making a transition where traditionally-managed infrastructure can no longer meet the demands of today's businesses. IT organizations are quickly adopting the public cloud, which is predominantly API-driven. To meet customer demands and save costs, application teams are architecting their applications to support a much higher level of elasticity, supporting technology like containers and public cloud resources.
These resources may only live for a matter of hours; therefore the traditional method of raising a ticket to request resources is no longer a viable option Pointing and clicking in a management console is NOT scale and increases the change of human error.

NEW QUESTION: 36
Consider the following Terraform 0.12 configuration snippet:
1. variable "vpc_cidrs" {
2. type = map
3. default = {
4. us-east-1 = "10.0.0.0/16"
5. us-east-2 = "10.1.0.0/16"
6. us-west-1 = "10.2.0.0/16"
7. us-west-2 = "10.3.0.0/16"
8. }
9. }

10.
11. resource "aws_vpc" "shared" {
12. cidr_block = _____
13. }
How would you define the cidr_block for us-east-1 in the aws_vpc resource using a variable?
A. var.vpc_cidrs[0]
B. var.vpc_cidrs["us-east-1"]
C. var.vpc_cidrs.0
D. vpc_cidrs["us-east-1"]
Answer: B (LEAVE A REPLY)

NEW QUESTION: 37
Which of the following challenges would Terraform be a candidate for solving? (Select THREE)
A. Have a single interoperable tool to manage the variety of services including GitHub repositories, MySQL database, and Kubernetes clusters.
B. Reduce the number of workflows needed for managing infrastructure across each of the companies public and private clouds.
C. Enable self-service infrastructure to allocate resources on your proprietary private cloud.
D. Utilize a single tool for all of the infrastructure and configuration management needs.
Answer: A,B,C (LEAVE A REPLY)

NEW QUESTION: 38
lookup retrieves the value of a single element from which of the below data type?
A. map
B. set
C. string
D. list
Answer: A (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/configuration/functions/lookup.html

NEW QUESTION: 39
Select the feature below that best completes the sentence:
The following list represents the different types of _____ available in Terraform.
1. max
2. min
3. join
4. replace
5. list

6. length
7. range

A. Backends
B. Data sources
C. Named values
D. Functions

Answer: D (LEAVE A REPLY)

Explanation

The Terraform language includes a number of built-in functions that you can call from within expressions to transform and combine values. The Terraform language does not support user-defined functions, and only the functions built into the language are available for use.

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

NEW QUESTION: 40

Which of the following are string functions? Select three

A. tostring
B. tonumber
C. Chomp
D. format
E. join

Answer: C,D,E (LEAVE A REPLY)

Explanation

tonumber and tostring are Type Conversion function

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

NEW QUESTION: 41

Which of the following is available only in Terraform Enterprise or Cloud workspaces and not in Terraform CLI?

A. Secure variable storage
B. Support for multiple cloud providers
C. Dry runs with terraform plan
D. Using the workspace as a data source

Answer: B (LEAVE A REPLY)

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

NEW QUESTION: 42

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: C (LEAVE A REPLY)
Explanation
Manually unlock the state for the defined configuration.
Reference: https://www.terraform.io/docs/cli/commands/force-unlock.html

NEW QUESTION: 43
A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?
A. terraform workspace dev
B. terraform workspace select dev
C. terraform workspace -switch dev
D. terraform workspace switch dev
Answer: B (LEAVE A REPLY)
Explanation
The terraform workspace select command is used to choose a different workspace to use for further operations.
https://www.terraform.io/docs/commands/workspace/select.html

NEW QUESTION: 44
Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.
A. Create individual workspaces for each team , and ask them to share modules across workspaces.
B. Implement a Private module registry in Terraform cloud , and ask teams to reference them.
C. Upgrade to Terraform enterprise , since this is not possible in terraform cloud.
D. Upload the modules in the terraform public module registry , and ask teams to reference them
Answer: B (LEAVE A REPLY)
Explanation
Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster. By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar. Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise. Terraform Cloud's private module registry helps you share Terraform modules across your organization.

https://www.terraform.io/docs/cloud/registry/index.html
https://www.terraform.io/docs/cloud/registry/publish.html

NEW QUESTION: 45
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: 46
Which of the following is considered a Terraform plugin?
A. Terraform language
B. Terraform tooling
C. Terraform logic
D. Terraform provider
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.
https://www.terraform.io/docs/plugins/basics.html

NEW QUESTION: 47
Only the user that generated a plan may apply it.
A. True
B. False
Answer: A (LEAVE A REPLY)
Explanation
The optional -out argument can be used to save the generated plan to a file for later execution with terraform apply, which can be useful when running Terraform in automation.
Reference: https://learn.hashicorp.com/tutorials/terraform/automate-terraform

NEW QUESTION: 48
Which of the following statements best describes the Terraform list(...) type?
A. a collection of values where each is identified by a string label.
B. a sequence of values identified by consecutive whole numbers starting with zero.
C. a collection of unique values that do not have any secondary identifiers or ordering.
D. a collection of named attributes that each have their own type.
Answer: B (LEAVE A REPLY)
Explanation
A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.
https://www.terraform.io/docs/configuration/types.html#structural-types

NEW QUESTION: 49
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)
Explanation
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: 50

In the example below, where is the value of the DNS record's IP address originating from?

```
1. resource "aws_route53_record" "www"
2. {
3. zone_id = aws_route53_zone.primary.zone_id
4. name = "www.example.com"
5. type = "A"
6. ttl = "300"
7. records = [module.web_server.instance_ip_address]
8. }
```

A. The regular expression named module.web_server
B. The output of a module named web_server
C. By querying the AWS EC2 API to retrieve the IP address
D. Value of the web_server parameter from the variables.tf file
Answer: (SHOW ANSWER)
Explanation
In a parent module, outputs of child modules are available in expressions as module.<MODULE NAME>.<OUTPUT NAME>.
For example, if a child module named web_server declared an output named instance_ip_address, you could access that value as module.web_server.instance_ip_address.

NEW QUESTION: 51
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: 52

True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

A. False

B. True

Answer: B (LEAVE A REPLY)

Explanation

The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

NEW QUESTION: 53

What command should you run to display all workspaces for the current configuration?

A. terraform workspace

B. terraform workspace show

C. terraform workspace list

D. terraform show workspace

Answer: C (LEAVE A REPLY)

Explanation

terraform workspace list

The command will list all existing workspaces.

Reference: https://www.terraform.io/docs/cli/commands/workspace/list.html

NEW QUESTION: 54

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: C,D,E (LEAVE A REPLY)

Explanation

Sentinel and Cost Estimation are also available in Terraform Cloud

https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION: 55

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

A. split

B. join

C. slice

D. chomp

Answer: D (LEAVE A REPLY)

Reference: https://www.terraform.io/docs/language/functions/chomp.html

NEW QUESTION: 56
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: A (LEAVE A REPLY)
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: 57
Terraform providers are always installed from the Internet.
A. True
B. False
Answer: (SHOW ANSWER)
Explanation
Terraform configurations must declare which providers they require, so that Terraform can install and use them.
Reference: https://www.terraform.io/docs/language/providers/configuration.html

NEW QUESTION: 58
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)
Explanation
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: 59
When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)
A. directly querying HashiCorp Vault for the secrets
B. command-line key/value pairs
C. use the -backend-config=PATH to specify a separate config file
D. interactively on the command line
Answer: B,C,D (LEAVE A REPLY)
Explanation
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.
With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments:
https://www.terraform.io/docs/backends/init.html#backend-initialization

NEW QUESTION: 60
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)
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: 61
What does terraform refresh command do?
A. terraform refresh syncs the state file with the real world infrastructure.
B. terraform refresh command basically updates the configuration file with the current state of the actual infrastructure
C. terraform refresh is use to change/modify the infrastructure based on the existing state file, at that moment.
D. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
E. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
Answer: (SHOW ANSWER)

NEW QUESTION: 62
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.
A. False
B. True
Answer: B (LEAVE A REPLY)

NEW QUESTION: 63
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. tainted and not destroyed
B. destroyed and recreated
C. destroyed and not recreated
D. recreated and tainted
Answer: B (LEAVE A REPLY)

NEW QUESTION: 64
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. data_ec2
B. track_tag
C. integration_ec2
D. digital_ec2
Answer: D (LEAVE A REPLY)

NEW QUESTION: 65
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)
Explanation
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: 66
Talal is a DevOps engineer and he has deployed the production infrastructure using Terraform. He is using a very large configuration file to maintain and update the actual infrastructure. As the infrastructure have grown to a very complex and large, he has started experiencing slowness when he run runs terraform plan. What are the options for him to resolve this slowness?
A. Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.
B. Run terraform refresh every time before running terraform plan.
C. Break large configurations into several smaller configurations that can each be independently applied.
D. Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.
Answer: C,D (LEAVE A REPLY)
Explanation
For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each

resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
Although 'Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.' is a solution, but its not always recommended. Instead of using -target as a means to operate on isolated portions of very large configurations, prefer instead to break large configurations into several smaller configurations that can each be independently applied. Data sources can be used to access information about resources created in other configurations, allowing a complex system architecture to be broken down into more manageable parts that can be updated independently.
Option 'Run terraform refresh every time before running terraform plan.' and 'Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.' is not correct because in both the cases terraform will query every resources of the infrastructure.

NEW QUESTION: 67
What is not processed when running a terraform refresh?
A. State file
B. Configuration file
C. Credentials
D. Cloud provider
Answer: C,D (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/cli/commands/refresh.html

NEW QUESTION: 68
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: 69
What does terrafom plan do ?
A. Create an execution plan by evaluating the difference between configuration file and state file.

B. Performs a refresh, unless explicitly disabled, and then apply the changes that are necessary to achieve the desired state specified in the configuration files.
C. Checks whether the execution plan for a set of changes matches your expectations by making changes to real resources or to the state.
D. Create an execution plan by evaluating the difference between configuration file and actual infrastructure.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 70
Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.
A. terraform state
B. terraform output
C. terraform validate
D. terraform plan
E. terraform apply
Answer: (SHOW ANSWER)

NEW QUESTION: 71
You have created two workspaces PROD and DEV. You have switched to DEV and provisioned DEV infrastructure from this workspace. Where is your state file stored?
A. terraform.d
B. terraform.tfstate
C. terraform.tfstate.DEV
D. terraform.tfstate.d
Answer: D (LEAVE A REPLY)
Explanation
Terraform stores the workspace states in a directory called terraform.tfstate.d. This directory should be treated similarly to default workspace state file terraform.tfstate main.tf provider.tf terraform.tfstate.d DEV terraform.tfstate # DEV workspace state file PROD terraform.tfstate # PROD workspace state file terraform.tfvars # Default workspace state file variables.tf

NEW QUESTION: 72
The following is a snippet from a Terraform configuration file:
Which, when validated, results in the following error:
Fill in the blank in the error message with the correct string from the list below.
A. version
B. multi
C. label
D. alias

Answer: D (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/configuration/providers.html#alias-multiple-providerinstances

NEW QUESTION: 73
Why might a user opt to include the following snippet in their configuration file?
A. Terraform 0.12 introduced substantial changes to the syntax used to write Terraform configuration
B. The user wants to ensure that the application being deployed is a minimum version of 0.12
C. this ensures that all Terraform providers are above a certain version to match the application being deployed
D. versions before Terraform 0.12 were not approved by HashiCorp to be used in production
Answer: A (LEAVE A REPLY)

NEW QUESTION: 74
What is the default backend for Terraform?
A. consul
B. gcs
C. local
D. etcd
Answer: (SHOW ANSWER)
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: 75
Which task does terraform init not perform?
A. Sources all providers present in the configuration and ensures they are downloaded and available locally
B. Connects to the backend
C. Sources any modules and copies the configuration locally
D. Validates all required variables are present
Answer: D (LEAVE A REPLY)
Reference: https://www.terraform.io/docs/cli/commands/init.html

NEW QUESTION: 76

When Terraform needs to be installed in a location where it does not have internet access to download the installer and upgrades, the installation is generally known as to be _____.

A. a private install
B. disconnected
C. air-gapped
D. non-traditional
Answer: (SHOW ANSWER)
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.

NEW QUESTION: 77
Which of the below features of Terraform can be used for managing small differences between different environments which can act more like completely separate working directories.

A. Repositories
B. Workspaces
C. Environment Variables
D. Backends
Answer: B (LEAVE A REPLY)
Explanation
workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. They are convenient in a number of situations, but cannot solve all problems.
A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. For example, a developer working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.

Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure.
When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.
https://www.terraform.io/docs/state/workspaces.html
https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces

NEW QUESTION: 78
Which of the following best describes the default local backend?
A. The local backend is where Terraform Enterprise stores logs to be processed by an log collector.
B. The local backend stores state on the local filesystem, locks the state using system APIs, and performs operations locally.
C. The local backend is the directory where resources deployed by Terraform have direct access to in order to update their current state.
D. The local backend is how Terraform connects to public cloud services, such as AWS, Azure, or GCP.
Answer: B (LEAVE A REPLY)
Explanation
The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.
terraform {
backend "local" {
path = "relative/path/to/terraform.tfstate"
}
}
https://www.terraform.io/docs/backends/types/local.html

NEW QUESTION: 79
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: 80
Terraform has detailed logs which can be enabled by setting the _____ environmental variable.
A. TF_TRACE
B. TF_DEBUG
C. TF_LOG
D. TF_INFO
Answer: (SHOW ANSWER)
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.
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION: 81
Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)
A. Unix
B. Red Hat
C. CentOS
D. Amazon Linux
E. Ubuntu
Answer: B,C,D,E (LEAVE A REPLY)
Explanation
https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements

NEW QUESTION: 82
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: (SHOW ANSWER)
Explanation
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: 83
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 => o.id }
C. [var.list[*].id]
D. [for o in var.list : o.id]
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 84
Which of the following variable definition files will terraform load automatically?
A. terraform.tfvar
B. Any files with names ending in .auto.tfvars.json
C. terraform.tfvars
D. terraform.tfvars.json
Answer: B,C,D (LEAVE A REPLY)
Explanation
Terraform also automatically loads a number of variable definitions files if they are present:
Files named exactly terraform.tfvars or terraform.tfvars.json.
Any files with names ending in .auto.tfvars or .auto.tfvars.json.
https://www.terraform.io/docs/configuration/variables.html
https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files

NEW QUESTION: 85

What is the standard workflow that a developer follows while working with terraform open source version?

A. Run terraform refresh to update the terraform state , then write the terraform code , and finally run terraform apply.

B. Run terraform destroy first since you need to start from fresh every time , before running terraform apply.

C. Write terraform code , and run terraform push , to update the terraform state to the remote repo , which in turn will take care of the next steps.

D. Write the terraform code on the developer machine , run terraform plan to check the changes , and run terraform apply to provision the infra.

Answer: D (LEAVE A REPLY)

Explanation

You do not need to run terraform refresh as terraform plan implicitly will run terraform refresh.

https://www.terraform.io/guides/core-workflow.html

NEW QUESTION: 86

Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)

A. Destroy

B. Apply

C. Import

D. Init

E. Validate

Answer: B,D (LEAVE A REPLY)

Reference: https://www.terraform.io/guides/core-workflow.html

NEW QUESTION: 87

Which feature of Terraform allows multiple state files for a single configuration file depending upon the environment?

A. Terraform Enterprise

B. Terraform Remote Backends

C. Terraform Modules

D. Terraform Workspaces

Answer: D (LEAVE A REPLY)

NEW QUESTION: 88

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: 89
Which of the below configuration file formats are supported by Terraform? (Select TWO)
A. Node
B. JSON
C. Go
D. YAML
E. HCL
Answer: B,E (LEAVE A REPLY)
Explanation
Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations.
https://www.terraform.io/docs/configuration/

NEW QUESTION: 90
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: 91
Terraform Enterprise (also referred to as pTFE) requires what type of backend database for a clustered deployment?
A. PostgreSQL

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

NEW QUESTION: 92
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: 93
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. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.
B. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
C. terraform apply will through an error.
D. The instance type will be changed to t2.micro and again will be changed to t2.medium
Answer: B (LEAVE A REPLY)

NEW QUESTION: 94
A variable az has the following default value. What will be the datatype of the variable?
az=["us-west-1a","us-east-1a"]
A. Object
B. String
C. List
D. Map
Answer: C (LEAVE A REPLY)

NEW QUESTION: 95
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 Functions
B. Terraform Backends
C. Dynamic Blocks
D. Terraform Expression
Answer: (SHOW ANSWER)

NEW QUESTION: 96
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)

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: 97

When writing Terraform code, HashiCorp recommends that you use how many spaces between each nesting level?

A. 0

B. 1

C. 2

D. 4

Answer: (SHOW ANSWER)

Explanation

The Terraform parser allows you some flexibility in how you lay out the elements in your configuration files, but the Terraform language also has some idiomatic style conventions which we recommend users always follow for consistency between files and modules written by different teams. Automatic source code formatting tools may apply these conventions automatically.

Indent two spaces for each nesting level.

When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs:

ami = "abc123"

instance_type = "t2.micro"

When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them. Use one blank line to separate the arguments from the blocks.

Use empty lines to separate logical groups of arguments within a block.

For blocks that contain both arguments and "meta-arguments" (as defined by the Terraform language semantics), list meta-arguments first and separate them from other arguments with one blank line. Place meta-argument blocks last and separate them from other blocks with one blank line.

resource "aws_instance" "example" {

```
count = 2 # meta-argument first
ami = "abc123"
instance_type = "t2.micro"
network_interface {
# ...
}
lifecycle { # meta-argument block last
create_before_destroy = true
}
}
```

Top-level blocks should always be separated from one another by one blank line. Nested blocks should also be separated by blank lines, except when grouping together related blocks of the same type (like multiple provisioner blocks in a resource).

Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example: root_block_device, ebs_block_device and ephemeral_block_device on aws_instance form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

NEW QUESTION: 98

In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?

A. Resource dependencies are identified and maintained in a file called resource.dependencies. Each terraform provider is required to maintain a list of all resource dependencies for the provider and it's included with the plugin during initialization when terraform init is executed. The file is located in the terraform.d folder.

B. The terraform binary contains a built-in reference map of all defined Terraform resource dependencies.

Updates to this dependency map are reflected in terraform versions. To ensure you are working with the latest resource dependency map you much be running the latest version of Terraform.

C. Resource dependencies are handled automatically by the depends_on meta_argument, which is set to true by default.

D. Terraform analyses any expressions within a resource block to find references to other objects, and treats those references as implicit ordering requirements when creating, updating, or destroying resources.

Answer: D (LEAVE A REPLY)

Explanation

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