



HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

NEW QUESTION 1

- (Exam Topic 1)

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

Explanation:

terraform workspace list

The command will list all existing workspaces.

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

NEW QUESTION 2

- (Exam Topic 1)

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: A

Explanation:

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

NEW QUESTION 3

- (Exam Topic 1)

What is the provider for this fictitious resource?

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

- A. vpc
- B. main
- C. aws
- D. test

Answer: C

Explanation:

Reference: <https://docs.aws.amazon.com/cloudformation-cli/latest/userguide/resource-types.html>

NEW QUESTION 4

- (Exam Topic 1)

You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully.

What will happen if you delete the VM using the cloud provider console, and run terraform apply again without changing any Terraform code?

- A. Terraform will remove the VM from state file
- B. Terraform will report an error
- C. Terraform will not make any changes
- D. Terraform will recreate the VM

Answer: D

NEW QUESTION 5

- (Exam Topic 1)

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 needs to install the necessary plugins first
- C. The Terraform CLI needs you to log into Terraform cloud first
- D. Terraform requires you to manually run terraform plan first

Answer: B

NEW QUESTION 6

- (Exam Topic 1)

Terraform provisioners can be added to any resource block.

- A. True
- B. False

Answer: A

Explanation:

<https://www.phillipsj.net/posts/introduction-to-terraform-provisioners/>

As you continue learning about Terraform, you will start hearing about provisioners. Terraform provisioners can be created on any resource and provide a way to execute actions on local or remote machines.

<https://www.terraform.io/language/resources/provisioners/local-exec>

NEW QUESTION 7

- (Exam Topic 1)

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

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

- A. dev
- B. azurerm_resource_group
- C. azurerm
- D. test

Answer: A

NEW QUESTION 8

- (Exam Topic 1)

You have used Terraform to create an ephemeral development environment in the cloud and are now ready to destroy all the infrastructure described by your Terraform configuration. To be safe, you would like to first see all the infrastructure that will be deleted by Terraform.

Which command should you use to show all of the resources that will be deleted? (Choose two.)

- A. Run terraform plan -destroy.
- B. This is not possible
- C. You can only show resources that will be created.
- D. Run terraform state rm *.
- E. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval.

Answer: AD

Explanation:

Reference: <https://www.terraform.io/docs/cli/commands/state/rm.html>

NEW QUESTION 9

- (Exam Topic 1)

Which argument(s) is (are) required when declaring a Terraform variable?

- A. type
- B. default
- C. description
- D. All of the above
- E. None of the above

Answer: B

Explanation:

The variable declaration can also include a default argument.

Reference: <https://www.terraform.io/docs/language/values/variables.html>

NEW QUESTION 10

- (Exam Topic 1)

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

Explanation:

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
Reference: <https://www.terraform.io/docs/cli/commands/taint.html>

NEW QUESTION 10

- (Exam Topic 1)

Terraform providers are always installed from the Internet.

- A. True
- B. False

Answer: B

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 14

- (Exam Topic 1)

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: B

Explanation:

"The terraform taint command informs Terraform that a particular object has become degraded or damaged. Terraform represents this by marking the object as "tainted" in the Terraform state, and Terraform will propose to replace it in the next plan you create." FYI - This command is deprecated. For Terraform v0.15.2 and later, we recommend using the -replace option with terraform apply instead. For Terraform v0.15.2 and later, we recommend using the -replace option with terraform apply to force Terraform to replace an object even though there are no configuration changes that would require it.
<https://www.terraform.io/cli/commands/taint>

NEW QUESTION 16

- (Exam Topic 1)

FILL BLANK

What is the name of the default file where Terraform stores the state?

Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

"This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment."
<https://www.terraform.io/language/state>

State

JUMP TO SECTION ▾

Terraform must store state about your managed infrastructure and configuration. This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures.

This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

NEW QUESTION 19

- (Exam Topic 1)

How can terraform plan aid in the development process?

- A. Validates your expectations against the execution plan without permanently modifying state
- B. Initializes your working directory containing your Terraform configuration files
- C. Formats your Terraform configuration files
- D. Reconciles Terraform's state against deployed resources and permanently modifies state using the current status of deployed resources

Answer: A

Explanation:

"The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. By default, when Terraform creates a plan it:

Reads the current state of any already-existing remote objects to make sure that the Terraform state is up-to-date.

Compares the current configuration to the prior state and noting any differences.

Proposes a set of change actions that should, if applied, make the remote objects match the configuration."

"The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review.

If Terraform detects that no changes are needed to resource instances or to root module output values, terraform plan will report that no actions need to be taken."

<https://www.terraform.io/cli/commands/plan>

NEW QUESTION 24

- (Exam Topic 1)

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: BD

Explanation:

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

NEW QUESTION 29

- (Exam Topic 1)

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

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

- A. True
- B. False

Answer: B

Explanation:

Unlike many other objects in the Terraform language, a provider block may be omitted if its contents would otherwise be empty. Terraform assumes an empty default configuration for any provider that is not explicitly configured. <https://www.terraform.io/language/providers/configuration>

NEW QUESTION 32

- (Exam Topic 1)

You're building a CI/CD (continuous integration/ continuous delivery) pipeline and need to inject sensitive variables into your Terraform run.

How can you do this safely?

- A. Pass variables to Terraform with a `--var` flag
- B. Copy the sensitive variables into your Terraform code
- C. Store the sensitive variables in a `secure_vars.tf` file
- D. Store the sensitive variables as plain text in a source code repository

Answer: A

Explanation:

<https://blog.gruntwork.io/a-comprehensive-guide-to-managing-secrets-in-your-terraform-code-1d586955ace1>

NEW QUESTION 35

- (Exam Topic 1)

Terraform provisioners that require authentication can use the _____ block.

- A. connection
- B. credentials
- C. secrets
- D. ssh

Answer: A

Explanation:

<https://www.terraform.io/language/resources/provisioners/connection>

"Most provisioners require access to the remote resource via SSH or WinRM and expect a nested connection block with details about how to connect."

"Connection blocks don't take a block label and can be nested within either a resource or a provisioner."

NEW QUESTION 39

- (Exam Topic 1)

You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (files). You need to enable debug messages to find this out.

Which of the following would achieve this?

- A. Set the environment variable TF_LOG=TRACE
- B. Set verbose logging for each provider in your Terraform configuration
- C. Set the environment variable TF_VAR_log=TRACE
- D. Set the environment variable TF_LOG_PATH

Answer: A

Explanation:

Although this will only output to stderr and if you need to review log file you will need to include TF_LOG_PATH=pathtofile
<https://www.terraform.io/internals/debugging>

NEW QUESTION 43

- (Exam Topic 1)

Which of the following is allowed as a Terraform variable name?

- A. count
- B. name
- C. source
- D. version

Answer: B

Explanation:

"The name of a variable can be any valid identifier except the following: source, version, providers, count, for_each, lifecycle, depends_on, locals."
<https://www.terraform.io/language/values/variables>

NEW QUESTION 44

- (Exam Topic 1)

One remote backend configuration always maps to a single remote workspace.

- A. True
- B. False

Answer: B

Explanation:

The remote backend can work with either a single remote Terraform Cloud workspace, or with multiple similarly-named remote workspaces (like networking-dev and networking-prod). The workspaces block of the backend configuration determines which mode it uses: To use a single remote Terraform Cloud workspace, set workspaces.name to the remote workspace's full name (like networking-prod). To use multiple remote workspaces, set workspaces.prefix to a prefix used in all of the desired remote workspace names. For example, set prefix = "networking-" to use Terraform cloud workspaces with names like networking-dev and networking-prod. This is helpful when mapping multiple Terraform CLI workspaces used in a single Terraform configuration to multiple Terraform Cloud workspaces.

NEW QUESTION 45

- (Exam Topic 1)

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

Explanation:

<https://www.terraform.io/language/state>

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 46

- (Exam Topic 1)

Which of the following is not a key principle of infrastructure as code?

- A. Versioned infrastructure
- B. Golden images
- C. Idempotence
- D. Self-describing infrastructure

Answer: B

Explanation:

Reference: <https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code#:~:text=Idempotence%20is%20a%20principle%20of,of%20the%20environment's%20starting%20state.>

NEW QUESTION 47

- (Exam Topic 1)

Your security team scanned some Terraform workspaces and found secrets stored in a plaintext in state files. How can you protect sensitive data stored in Terraform state files?

- A. Delete the state file every time you run Terraform
- B. Store the state in an encrypted backend
- C. Edit your state file to scrub out the sensitive data
- D. Always store your secrets in a secrets.tfvars file.

Answer: B

NEW QUESTION 48

- (Exam Topic 1)

Terraform can import modules from a number of sources – which of the following is not a valid source?

- A. FTP server
- B. GitHub repository
- C. Local path
- D. Terraform Module Registry

Answer: A

Explanation:

<https://www.terraform.io/language/modules/sources>

NEW QUESTION 53

- (Exam Topic 1)

Which of the following is the correct way to pass the value in the variable num_servers into a module with the input servers?

- A. servers = num_servers
- B. servers = variable.num_servers
- C. servers = var(num_servers)
- D. servers = var.num_servers

Answer: D

Explanation:

"Within the module that declared a variable, its value can be accessed from within expressions as var.<NAME>, where <NAME> matches the label given in the declaration block:

Note: Input variables are created by a variable block, but you reference them as attributes on an object named var."

<https://www.terraform.io/language/values/variables#using-input-variable-values>

NEW QUESTION 54

- (Exam Topic 1)

What type of block is used to construct a collection of nested configuration blocks?

- A. for_each
- B. repeated
- C. nesting
- D. dynamic

Answer: D

Explanation:

<https://www.terraform.io/language/expressions/dynamic-blocks>

NEW QUESTION 59

- (Exam Topic 1)

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

- A. True
- B. False

Answer: A

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 64

- (Exam Topic 1)

How would you reference the "name" value of the second instance of this fictitious resource?

```
resource "aws_instance" "web" {  
  count = 2  
  name = "terraform-${count.index}"  
}
```

- 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: B

Explanation:

<https://www.terraform.io/language/meta-arguments/count#referring-to-instances> Reference: <https://www.terraform.io/docs/configuration-0-11/interpolation.html>

NEW QUESTION 67

- (Exam Topic 1)

You have never used Terraform before and would like to test it out using a shared team account for a cloud provider. The shared team account already contains 15 virtual machines (VM). You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully. What should you do to delete the newly-created VM with Terraform?

- A. The Terraform state file contains all 16 VMs in the team account
- B. Execute terraform destroy and select the newly-created VM.
- C. The Terraform state file only contains the one new V
- D. Execute terraform destroy.
- E. Delete the Terraform state file and execute Terraform apply.
- F. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.

Answer: B

Explanation:

You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully. read the question carefully "Terraform configuration containing one VM, perform terraform apply" so only one VM is in state file.

NEW QUESTION 68

- (Exam Topic 1)

When does terraform apply reflect changes in the cloud environment?

- A. Immediately
- B. However long it takes the resource provider to fulfill the request
- C. After updating the state file
- D. Based on the value provided to the -refresh command line argument
- E. None of the above

Answer: B

NEW QUESTION 71

- (Exam Topic 1)

HashiCorp Configuration Language (HCL) supports user-defined functions.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/language/functions>

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

NEW QUESTION 72

- (Exam Topic 1)

What is not processed when running a terraform refresh?

- A. State file
- B. Configuration file
- C. Credentials
- D. Cloud provider

Answer: B

Explanation:

"The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match."

NEW QUESTION 76

- (Exam Topic 1)

A Terraform provisioner must be nested inside a resource configuration block.

- A. True
- B. False

Answer: A

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 79

- (Exam Topic 1)

If writing Terraform code that adheres to the Terraform style conventions, how would you properly indent each nesting level compared to the one above it?

- A. With four spaces
- B. With a tab
- C. With three spaces
- D. With two spaces

Answer: D

Explanation:

<https://www.terraform.io/language/syntax/style#style-conventions>

NEW QUESTION 84

- (Exam Topic 1)

Setting the TF_LOG environment variable to DEBUG causes debug messages to be logged into syslog.

- A. True
- B. False

Answer: B

Explanation:

TF_LOG_PATH IS NOT REQUIRED, in the docs, they do not mention HAVE TO SET TF_LOG_PATH, it is optional, therefore without TF_LOG_PATH will cause detailed logs to appear on stderr.

<https://www.computerhope.com/jargon/s/stderr.htm#:~:text=Stderr%2C%20also%20known%20as%20standard,>

NEW QUESTION 86

- (Exam Topic 1)

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

Explanation:

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 87

- (Exam Topic 1)

When using Terraform to deploy resources into Azure, which scenarios are true regarding state files? (Choose two.)

- A. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in a new state file
- B. When a change is made to the resources via the Azure Cloud Console, Terraform will update the state file to reflect them during the next plan or apply
- C. When a change is made to the resources via the Azure Cloud Console, the current state file will not be updated
- D. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in the current state file

Answer: BC

NEW QUESTION 92

- (Exam Topic 1)

Which of the following is not a valid Terraform collection type?

- A. list
- B. map
- C. tree
- D. set

Answer: C

Explanation:

<https://www.terraform.io/language/expressions/type-constraints#collection-types>

NEW QUESTION 93

- (Exam Topic 1)

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: D

NEW QUESTION 98

- (Exam Topic 1)

Which of the following is not true of Terraform providers?

- A. Providers can be written by individuals
- B. Providers can be maintained by a community of users
- C. Some providers are maintained by HashiCorp
- D. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers
- E. None of the above

Answer: E

Explanation:

<https://registry.terraform.io/providers/hashicorp/google/latest> - This provider is collaboratively maintained by the Google Terraform Team at Google and the Terraform team at HashiCorp

<https://www.terraform.io/language/providers>

NEW QUESTION 100

- (Exam Topic 1)

FILL BLANK

You need to specify a dependency manually.

What resource meta-parameter can you use to make sure Terraform respects the dependency?

Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Reference: <https://www.terraform.io/docs/language/functions/file.html>

NEW QUESTION 102

- (Exam Topic 1)

How can you trigger a run in a Terraform Cloud workspace that is connected to a Version Control System (VCS) repository?

- A. Only Terraform Cloud organization owners can set workspace variables on VCS connected workspaces
- B. Commit a change to the VCS working directory and branch that the Terraform Cloud workspace is connected to
- C. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces
- D. Only Terraform Cloud organization owners can approve plans in VCS connected workspaces

Answer: B

Explanation:

"In a workspace linked to a VCS repository, runs start automatically when you merge or commit changes to version control.

A workspace is linked to one branch of a VCS repository and ignores changes to other branches. You can specify which files and directories within your repository trigger runs. "

<https://www.terraform.io/cloud-docs/run/ui#automatically-starting-runs>

NEW QUESTION 103

- (Exam Topic 1)

Which of the following is not an action performed by terraform init?

- A. Create a sample main.tf file
- B. Initialize a configured backend
- C. Retrieve the source code for all referenced modules
- D. Load required provider plugins

Answer: A

NEW QUESTION 106

- (Exam Topic 1)

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

Explanation:

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 108

- (Exam Topic 1)

Your DevOps team is currently using the local backend for your Terraform configuration. You would like to move to a remote backend to begin storing the state file in a central location. Which of the following backends would not work?

- A. Amazon S3
- B. Artifactory
- C. Git
- D. Terraform Cloud

Answer: C

Explanation:

<https://www.terraform.io/cdktf/concepts/remote-backends> https://docs.gitlab.com/ee/user/infrastructure/iac/terraform_state.html

NEW QUESTION 109

- (Exam Topic 1)

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

Explanation:

Backends define where Terraform's state snapshots are stored. A given Terraform configuration can either specify a backend, integrate with Terraform Cloud, or do neither and default to storing state locally.

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 114

- (Exam Topic 1)

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: A

Explanation:

<https://www.terraform.io/cli/commands/fmt>

NEW QUESTION 117

- (Exam Topic 2)

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

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 122

- (Exam Topic 2)

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

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 125

- (Exam Topic 2)

True or False: A list(...) contain a number of values of the same type while an object(...) can contain a number of values of different types.

- A. False
- B. True

Answer: B

Explanation:

Collection Types

A collection type allows multiple values of one other type to be grouped together as a single value. The type of value within a collection is called its element type.

All collection types must have an element type, which is provided as the argument to their constructor.

For example, the type `list(string)` means "list of strings", which is a different type than `list(number)`, a list of numbers. All elements of a collection must always be of the same type.

The three kinds of collection type in the Terraform language are:

* `list(...)`: a sequence of values identified by consecutive whole numbers starting with zero.

The keyword `list` is a shorthand for `list(any)`, which accepts any element type as long as every element is the same type. This is for compatibility with older configurations; for new code, we recommend using the full form.

* `map(...)`: a collection of values where each is identified by a string label.

The keyword `map` is a shorthand for `map(any)`, which accepts any element type as long as every element is the same type. This is for compatibility with older configurations; for new code, we recommend using the full form.

* `set(...)`: a collection of unique values that do not have any secondary identifiers or ordering. <https://www.terraform.io/docs/configuration/types.html>

Structural Types

A structural type allows multiple values of several distinct types to be grouped together as a single value. Structural types require a schema as an argument, to specify which types are allowed for which elements.

The two kinds of structural type in the Terraform language are:

* `object(...)`: a collection of named attributes that each have their own type.

The schema for object types is `{ <KEY> = <TYPE>, <KEY> = <TYPE>, ... }` — a pair of curly braces containing a comma-separated series of `<KEY> = <TYPE>` pairs. Values that match the object type must contain all of the specified keys, and the value for each key must match its specified type. (Values with additional keys can still match an object type, but the extra attributes are discarded during type conversion.)

* `tuple(...)`: a sequence of elements identified by consecutive whole numbers starting with zero, where each element has its own type.

The schema for tuple types is `[<TYPE>, <TYPE>, ...]` — a pair of square brackets containing a comma-separated series of types. Values that match the tuple type must have exactly the same number of elements (no more and no fewer), and the value in each position must match the specified type for that position.

For example: an object type of `object({ name=string, age=number })` would match a value like the following:

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

Also, an object type of `object({ id=string, cidr_block=string })` would match the object produced by a reference to an `aws_vpc` resource, like `aws_vpc.example_vpc`; although the resource has additional attributes, they would be discarded during type conversion.

Finally, a tuple type of `tuple([string, number, bool])` would match a value like the following: `["a", 15, true]`

<https://www.terraform.io/docs/configuration/types.html>

NEW QUESTION 130

- (Exam Topic 2)

Environment variables can be used to set variables. The environment variables must be in the format "`__<variablename>`". Select the correct prefix string from the following list.

- A. `TF_CLI_ARGS`
- B. `TF_VAR`
- C. `TF_VAR_`
- D. `TF_VAR_ENV`

Answer: C

Explanation:

Environment variables can be used to set variables. The environment variables must be in the format `TF_VAR_name` and this will be checked last for a value. For example:

```
export TF_VAR_region=us-west-1
```

```
export TF_VAR_ami=ami-049d8641 export TF_VAR_alist='[1,2,3]'
```

```
export TF_VAR_amap='{ foo = "bar", baz = "qux" }'
```

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

NEW QUESTION 131

- (Exam Topic 2)

You want to use different AML images for different regions and for the purpose you have defined following code block.

```
* 1.variable "images"  
* 2.{  
* 3. type = "map"  
* 4.  
* 5. default = {  
* 6. us-east-1 = "image-1234"  
* 7. us-west-2 = "image-4567"  
* 8. us-west-1 = "image-4589"  
* 9. }  
* 10.}
```

What of the following approaches needs to be followed in order to select image-4589?

- A. var.images["us-west-1"]
- B. var.images[3]
- C. var.images[2]
- D. lookup(var.images["us-west-1"])

Answer: A

NEW QUESTION 136

- (Exam Topic 2)

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

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 137

- (Exam Topic 2)

Terraform has detailed logs which can be enabled by setting the _____ environmental variable.

- A. TF_TRACE
- B. TF_DEBUG
- C. TF_LOG
- D. TF_INFO

Answer: C

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 141

- (Exam Topic 2)

Which of the following clouds does not have a provider maintained HashiCorp?

- A. IBM Cloud
- B. DigitalOcean
- C. OpenStack
- D. AWS

Answer: A

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 144

- (Exam Topic 2)

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

- A. Yes
- B. No

Answer: B

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 147

- (Exam Topic 2)

You have created 2 workspaces PROD and RQA. You have switched to RQA and provisioned RQA infrastructure from this workspace. Where is your state file stored?

- A. terraform.tfstate.d
- B. terraform.d
- C. terraform.tfstate.RQA
- D. terraform.tfstate

Answer: A

NEW QUESTION 150

- (Exam Topic 2)

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

Explanation:

<https://www.terraform.io/docs/configuration/functions/lookup.html>

NEW QUESTION 152

- (Exam Topic 2)

Terraform init can indeed be run only a few times, because, every time terraform init will initialize the project , and download all plugins from the internet repository , regardless of whether they were present or not , and this increases the waiting time

- A. True
- B. False

Answer: B

Explanation:

Re-running init with modules already installed will install the sources for any modules that were added to configuration since the last init, but will not change any already-installed modules. Use -upgrade to override this behavior, updating all modules to the latest available source code.

<https://www.terraform.io/docs/commands/init.html>

NEW QUESTION 154

- (Exam Topic 2)

What is the default backend for Terraform?

- A. consul
- B. gcs
- C. local
- D. etcd

Answer: C

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 157

- (Exam Topic 2)

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

Explanation:

<https://www.terraform.io/docs/configuration/variables.html>

NEW QUESTION 158

- (Exam Topic 2)

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: ABD

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 160

- (Exam Topic 2)

terraform refresh command will not modify infrastructure, but does modify the state file.

- A. True
- B. False

Answer: A

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 164

- (Exam Topic 2)

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

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 166

- (Exam Topic 2)

By default, a defined provisioner is a creation-time provisioner.

- A. True
- B. False

Answer: A

Explanation:

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

NEW QUESTION 167

- (Exam Topic 2)

Please identify the offerings which are unique to Terraform Enterprise, and not available in either Terraform OSS, or Terraform Cloud. Select four.

- A. Audit Logs
- B. Private Network Connectivity
- C. VCS Integration
- D. Sentinel
- E. Clustering

Answer: ABE

Explanation:

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

NEW QUESTION 172

- (Exam Topic 2)

Which of the following best describes a Terraform provider?

- A. A plugin that Terraform uses to translate the API interactions with the service or provider.

- B. Serves as a parameter for a Terraform module that allows a module to be customized.
- C. Describes an infrastructure object, such as a virtual network, compute instance, or other components.
- D. A container for multiple resources that are used together.

Answer: A

Explanation:

A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g. Terraform Cloud, DNSimple, Cloudflare).
<https://www.terraform.io/docs/providers/index.html>

NEW QUESTION 174

- (Exam Topic 2)

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: CD

Explanation:

<https://www.terraform.io/docs/commands/refresh.html>

NEW QUESTION 175

- (Exam Topic 2)

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 _____.

- A. First Time Configuration
- B. Default Configuration
- C. Changing Configuration
- D. Partial Configuration
- E. Incomplete Configuration

Answer: D

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:

* Interactively: Terraform will interactively ask you for the required values, unless interactive input is disabled. Terraform will not prompt for optional values.

* File: A configuration file may be specified via the init command line. To specify a file, use the -backend-config=PATH option when running terraform init. If the file contains secrets it may be kept in a secure data store, such as Vault, in which case it must be downloaded to the local disk before running Terraform.

* Command-line key/value pairs: Key/value pairs can be specified via the init command line. Note that many shells retain command-line flags in a history file, so this isn't recommended for secrets. To specify a single key/value pair, use the -backend-config="KEY=VALUE" option when running terraform init.

<https://www.terraform.io/docs/backends/config.html#partial-configuration>

NEW QUESTION 177

- (Exam Topic 2)

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: AC

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 179

- (Exam Topic 3)

You have been given requirements to create a security group for a new application. Since your organization standardizes on Terraform, you want to add this new security group with the fewest number of lines of code. What feature could you use to iterate over a list of required tcp ports to add to the new security group?

- A. dynamic backend
- B. splat expression
- C. terraform import
- D. dynamic block

Answer: D

Explanation:

A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value and generates a nested block for each element of that complex value.

<https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks>

NEW QUESTION 182

- (Exam Topic 3)

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

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 185

- (Exam Topic 3)

In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

- A. 5
- B. 2
- C. 3
- D. 1

Answer: D

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 190

- (Exam Topic 3)

Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?

- A. False
- B. True

Answer: A

Explanation:

Terraform module need not be public and open-source. Module can be placed in

- * Local paths
- * Terraform Registry
- * GitHub
- * Bitbucket
- * Generic Git, Mercurial repositories
- * HTTP URLs
- * S3 buckets
- * GCS buckets <https://www.terraform.io/docs/modules/sources.html>

NEW QUESTION 194

- (Exam Topic 3)

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

Explanation:

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 197

- (Exam Topic 3)

Which of the following challenges would Terraform be a candidate for solving? (Select THREE)

- A. Enable self-service infrastructure to allocate resources on your proprietary private cloud.
- B. Reduce the number of workflows needed for managing infrastructure across each of the companies public and private clouds.
- C. Utilize a single tool for all of the infrastructure and configuration management needs.
- D. Have a single interoperable tool to manage the variety of services including GitHub repositories, MySQL database, and Kubernetes clusters.

Answer: ABD

NEW QUESTION 202

- (Exam Topic 3)

The canonical format may change in minor ways between Terraform versions, so after upgrading Terraform it is recommended to proactively run.

- A. terraform fmt
- B. terraform init
- C. terraform validate
- D. terraform plan

Answer: A

NEW QUESTION 207

- (Exam Topic 3)

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

- A. terraform taint
- B. terraform apply
- C. terraform graph
- D. terraform refresh

Answer: A

Explanation:

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

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

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

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

This example will taint a single resource:

```
$ terraform taint aws_security_group.allow_all
```

The resource aws_security_group.allow_all in the module root has been marked as tainted. <https://www.terraform.io/docs/commands/taint.html>

NEW QUESTION 211

- (Exam Topic 3)

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

Explanation:

<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 215

- (Exam Topic 3)

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 hi
- C. Then there will be no chance of any inconsistencies.
- D. 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.
- E. Enable terraform state locking for the S3 backend using DynamoDB tabl
- F. This prevents others from acquiring the lock and potentially corrupting your state.

Answer: D

Explanation:

S3 backend support state locking using DynamoDB. <https://www.terraform.io/docs/state/locking.html>

NEW QUESTION 218

- (Exam Topic 3)

State locking does not happen automatically and must be specified at run

- A. False
- B. True

Answer: A

Explanation:

State locking happens automatically on all operations that could write state. <https://www.terraform.io/docs/state/locking.html>

NEW QUESTION 219

- (Exam Topic 3)

```
* 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 dependenc
- C. Nothing else needs to be done from your end.
- 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.
- E. 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

Explanation:

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

NEW QUESTION 220

- (Exam Topic 3)

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

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 222

- (Exam Topic 3)

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: C

Explanation:

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

NEW QUESTION 225

- (Exam Topic 3)

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: ACEF

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 230

- (Exam Topic 3)

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

Explanation:

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 235

- (Exam Topic 3)

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: AC

Explanation:

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 237

- (Exam Topic 3)

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

Explanation:

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 238

- (Exam Topic 3)

Which flag would be used within a Terraform configuration block to identify the specific version of a provider required?

- A. required-provider
- B. required-version
- C. required_providers
- D. required_versions

Answer: C

Explanation:

For production use, you should constrain the acceptable provider versions via configuration file to ensure that new versions with breaking changes will not be automatically installed by terraform init in the future.

```
Example terraform {  
  required_providers { aws = ">= 2.7.0"  
}  
}
```

NEW QUESTION 239

- (Exam Topic 3)

When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?

- A. Object Storage
- B. State Locking
- C. WorkSpaces
- D. Encryption

Answer: B

Explanation:

If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state.

State locking happens automatically on all operations that could write state. You won't see any message that it is happening. If state locking fails, Terraform will not continue. You can disable state locking for most commands with the -lock flag but it is not recommended.

If acquiring the lock is taking longer than expected, Terraform will output a status message. If Terraform doesn't output a message, state locking is still occurring if your backend supports it.

Not all backends support locking. Please view the list of backend types for details on whether a backend supports locking or not.

<https://www.terraform.io/docs/state/locking.html>

NEW QUESTION 242

- (Exam Topic 3)

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: BCD

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 246

- (Exam Topic 3)

A data block requests that Terraform read from a given data source and export the result under the given local name.

- A. False
- B. True

Answer: B

NEW QUESTION 247

- (Exam Topic 3)

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: D

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 248

- (Exam Topic 3)

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

Explanation:

Check resource addressing <https://www.terraform.io/docs/internals/resource-addressing.html>

NEW QUESTION 250

- (Exam Topic 3)

Terraform Enterprise currently supports running under which the following operating systems?

- A. Ubuntu
- B. Amazon Linux
- C. Debian
- D. CentOS
- E. Red Hat Enterprise Linux
- F. Oracle Linux

Answer: ABCDEF

Explanation:

Terraform Enterprise runs on Linux instances, and you must prepare a running Linux instance for Terraform Enterprise before running the installer. You will start and manage this instance like any other server.

Terraform Enterprise currently supports running under the following operating systems: Standalone deployment:

Debian 7.7+

Ubuntu 14.04.5 / 16.04 / 18.04

Red Hat Enterprise Linux 7.4 - 7.8 CentOS 6.x / 7.4 - 7.8

Amazon Linux 2014.03 / 2014.09 / 2015.03 / 2015.09 / 2016.03 / 2016.09 / 2017.03 / 2017.09 / 2018.03 / 2.0

Oracle Linux 7.4 - 7.8 <https://www.terraform.io/docs/enterprise/before-installing/index.html>

NEW QUESTION 253

- (Exam Topic 3)

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. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
- B. The instance type will be changed to t2.micro and again will be changed to t2.medium
- C. terraform apply will through an error.
- D. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.

Answer: A

NEW QUESTION 255

- (Exam Topic 3)

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

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 259

- (Exam Topic 3)

You can migrate the Terraform backend but only if there are no resources currently being managed.

- A. False
- B. True

Answer: A

Explanation:

If you need to migrate to another backend, such as Terraform Cloud, so you can continue managing it. By migrating your Terraform state, you can hand off infrastructure without de-provisioning anything.
<https://www.terraform.io/docs/cloud/migrate/index.html>

NEW QUESTION 261

- (Exam Topic 3)

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

Explanation:

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

NEW QUESTION 262

- (Exam Topic 3)

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

Explanation:

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

NEW QUESTION 264

- (Exam Topic 3)

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

Explanation:

<https://www.terraform.io/docs/state/remote.html>

NEW QUESTION 269

- (Exam Topic 3)

What does terraform refresh command do?

- A. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
- 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 syncs the state file with the real world infrastructure.

Answer: E

NEW QUESTION 272

- (Exam Topic 3)

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

Explanation:

The terraform workspace new command is used to create a new workspace. <https://www.terraform.io/docs/commands/workspace/new.html>

NEW QUESTION 276

- (Exam Topic 3)

You have provisioned some aws resources in your test environment through Terraform for a POC work. After the POC, now you want to destroy the resources but before destroying them you want to check what resources will be getting destroyed through terraform. what are the options of doing that? (Select TWO)

- A. Use terraform destroy command
- B. This is not possible
- C. Use terraform plan command
- D. Use terraform plan -destroy command.

Answer: AD

Explanation:

<https://learn.hashicorp.com/terraform/getting-started/destroy>

NEW QUESTION 278

- (Exam Topic 3)

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. integration_ec2
- B. digital_ec2
- C. track_tag
- D. data_ec2

Answer: B

NEW QUESTION 283

- (Exam Topic 4)

A Terraform output that sets the "sensitive" argument to true will not store that value in the state file.

- A. True
- B. False

Answer: B

Explanation:

Reference: <https://www.terraform.io/language/values/outputs>

NEW QUESTION 286

- (Exam Topic 4)

You have decided to create a new Terraform workspace to deploy a development environment. What is different about this workspace?

- A. It uses a different branch of code It uses a different backend
- B. It has its own state file
- C. It pulls in a different terraform.tvvars file

Answer: C

NEW QUESTION 287

- (Exam Topic 4)

How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? (Choose two.)

- A. A full audit trail of the request and fulfillment process is generated
- B. A request must be submitted for infrastructure changes
- C. As additional resources are required, more tickets are submitted
- D. A catalog of approved resources can be accessed from drop down lists in a request form

Answer: BC

NEW QUESTION 289

- (Exam Topic 4)

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: ACF

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 294

- (Exam Topic 4)

What is the result of the following terraform function call?

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 298

- (Exam Topic 4)

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

- A. source = "git::https://example.com/my_test_module.git@tag=v1.0.4"
- B. source = "git::https://example.com/my_test_module.git&ref=v1.0.4"
- C. source = "git::https://example.com/my_test_module.git#tag=v1.0.4"
- D. source = "git::https://example.com/my_test_module.git?ref=v1.0.4"

Answer: D

Explanation:

<https://www.terraform.io/docs/modules/sources.html#selecting-a-revision>

NEW QUESTION 303

- (Exam Topic 4)

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

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 307

- (Exam Topic 4)

You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name the instance resource "main".

Which of these is the correct way to define the output value using HCL2?

A.

```
output "instance_ip_addr" {  
    value = "${aws_instance.main.private_ip}"  
}
```

B.

```
output "instance_ip_addr" {  
    return aws_instance.main.private_ip  
}
```

A. Option A

B. Option B

Answer: A

NEW QUESTION 308

- (Exam Topic 4)

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

Answer: A

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.

NEW QUESTION 310

- (Exam Topic 4)

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

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 315

- (Exam Topic 4)

True or False. The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. If drift is detected between the real-world infrastructure and the last known-state, it will modify the infrastructure to correct the drift.

A. False

B. True

Answer: A

Explanation:

<https://www.terraform.io/docs/commands/refresh.html>

NEW QUESTION 317

- (Exam Topic 4)

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

Explanation:

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

NEW QUESTION 320

- (Exam Topic 4)

Terraform console provides an interactive command-line console for evaluating and experimenting with expressions. You can use it to test interpolations before using them in configurations and to interact with any values currently saved in state.

Which configuration consistency errors does terraform validate report?

- A. A mix of spaces and tabs in configuration files
- B. Differences between local and remote state
- C. Terraform module isn't the latest version
- D. Declaring a resource identifier more than once

Answer: D

Explanation:

validate will look for syntax errors "Declaring a resource identifier more than once" is a syntax error

NEW QUESTION 324

- (Exam Topic 4)

Terra form installs its providers during which phase?

- A. Man
- B. Init
- C. Refresh
- D. All of the above

Answer: B

Explanation:

Providers are installed in the init phase

NEW QUESTION 326

- (Exam Topic 4)

Any user can publish modules to the public Terraform Module Registry.

- A. True
- B. False

Answer: B

NEW QUESTION 327

- (Exam Topic 4)

You have modified your Terraform configuration to fix a typo in the Terraform ID of a resource from aws_security_group.http to aws_security_group.http

Original configuration:

```
resource "aws_security_group" "htp" {  
  name = "http"  
  ingress {  
    from_port = "80"  
    to_port   = "80"  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

Updated configuration:

```
resource "aws_security_group" "http" {  
  name = "http"  
  ingress {  
    from_port = "80"  
    to_port   = "80"  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

Which of the following commands would you run to update the ID in state without destroying the resource?

- A. terraform refresh
- B. terraform apply
- C. terraform mv aws-security-group.htp aws-security-group.http

Answer: C

Explanation:

The terraform state mv command changes which resource address in your configuration is associated with a particular real-world object. Use this to preserve an object when renaming a resource, or when moving a resource into or out of a child module.

NEW QUESTION 328

- (Exam Topic 4)

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: ABC

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 331

- (Exam Topic 4)

Which of the following arguments are required when declaring a Terraform output?

- A. sensitive
- B. description
- C. default
- D. value

Answer: D

NEW QUESTION 334

- (Exam Topic 4)

Which of the following is not a benefit of adopting infrastructure as code?

- A. Automation
- B. Versioning
- C. Reusability of code
- D. Interpolation

Answer: D

NEW QUESTION 335

- (Exam Topic 4)

Running terraform fmt without any flags in a directory with Terraform configuration files will check the formatting of those files without changing their contents.

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 336

- (Exam Topic 4)

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

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 341

- (Exam Topic 4)

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

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 343

- (Exam Topic 4)

True or False? terraform init cannot automatically download Community providers.

- A. False
- B. True

Answer: B

NEW QUESTION 344

- (Exam Topic 4)

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

- A. Run terraform apply -upgrade
- B. Run terraform init -upgrade
- C. Run terraform refresh
- D. Upgrade your version of Terraform

Answer: B

Explanation:

[-upgrade] - Opt to upgrade modules and plugins as part of their respective installation steps. See the sections below for more details. Reference:

<https://www.terraform.io/cli/commands/init#upgrade>

NEW QUESTION 347

- (Exam Topic 4)

From the code below, identify the implicit dependency:

- A. The EIP with an id of ami-2757f631
- B. The AMI used for the EC2 instance
- C. The EC2 instance labeled web_server
- D. The S3 bucket labeled company_data

Answer: C

NEW QUESTION 350

- (Exam Topic 4)

Multiple provider instances blocks for AWS can be part of a single configuration file?

- A. False
- B. True

Answer: B

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 352

- (Exam Topic 4)

Which of the following locations can Terraform use as a private source for modules? (Choose two.)

- A. Internally hosted SCM (Source Control Manager) platform
- B. Public Terraform Module Registry
- C. Private repository on GitHub
- D. Public repository on GitHub

Answer: AC

NEW QUESTION 356

- (Exam Topic 4)

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: CD

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 358

- (Exam Topic 4)

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

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 360

- (Exam Topic 4)

What resource dependency information is stored in Terraform's state?

- A. Only implicit dependencies are stored in state.
- B. Both implicit and explicit dependencies are stored in state.
- C. Only explicit dependencies are stored in state.
- D. No dependency information is stored in state.

Answer: B

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 362

- (Exam Topic 4)

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

- A. True
- B. False

Answer: A

Explanation:

Initialization

Whenever a configuration's backend changes, you must run terraform init again to validate and configure the backend before you can perform any plans, applies, or state operations.

When changing backends, Terraform will give you the option to migrate your state to the new backend. This lets you adopt backends without losing any existing state.

To be extra careful, we always recommend manually backing up your state as well. You can do this by simply copying your terraform.tfstate file to another location.

The initialization process should create a backup as well, but it never hurts to be safe!

<https://www.terraform.io/language/settings/backends/configuration>

NEW QUESTION 366

- (Exam Topic 4)

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: CDE

Explanation:

<https://www.terraform.io/docs/registry/modules/publish.html#requirements>

NEW QUESTION 369

- (Exam Topic 4)

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

NEW QUESTION 374

- (Exam Topic 4)

What is the result of the following terraform function call?

- A. hello
- B. what?
- C. goodbye

Answer: B

Explanation:

<https://www.terraform.io/docs/configuration/functions/lookup.html>

NEW QUESTION 376

- (Exam Topic 4)

All modules published on the official Terraform Module Registry have been verified by HashiCorp.

- A. True
- B. False

Answer: B

Explanation:

<https://registry.terraform.io/>

Only modules considered "Verified Modules" are reviewed by Hashicorp, otherwise anyone can publish modules on the Terraform Registry.

Reference: <https://www.terraform.io/registry/modules/verified> <https://www.terraform.io/registry/modules/publish>

NEW QUESTION 380

- (Exam Topic 4)

Your team uses terraform OSS . You have created a number of reusable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option/way to do that?

- A. Terraform modules cannot be shared in OSS version . Each developer needs to maintain their own modules and leverage them in the main tf file.
- B. Upload your modules with proper versioning in the terraform public module registry . Terraform OSS is directly integrated with the public module registry , and can reference the modules from the code in the main tf file.
- C. Terraform module sharing is only available in Enterprise version via terraform private module registry , so no way to enable it in OSS version.
- D. Store your modules in a NAS/ shared file server , and ask your team members to directly reference the code from there
- E. This is the only viable option in terraform OSS ,which is better than individually maintaining module versions for every developer.

Answer: B

Explanation:

Software development encourages code reuse through reusable artifacts, such as libraries, packages and modules. Most programming languages enable developers to package and publish these reusable components and make them available on a registry or feed. For example, Python has Python Package Index and PowerShell has PowerShell Gallery.

For Terraform users, the Terraform Registry enables the distribution of Terraform modules, which are reusable configurations. The Terraform Registry acts as a centralized repository for module sharing, making modules easier to discover and reuse.

The Registry is available in two variants:

- * Public Registry houses official Terraform providers -- which are services that interact with an API to expose and manage a specific resource -- and community-contributed modules.

- * Private Registry is available as part of the Terraform Cloud, and can host modules internally within an organization.

<https://www.terraform.io/docs/registry/index.html>

NEW QUESTION 384

- (Exam Topic 4)

How would you reference the Volume IDs associated with the ebs_block_device blocks in this configuration?

```
resource "aws_instance" "example" {
  ami = "ami-abc123"
  instance_type = "t2.micro"

  ebs_block_device {
    device_name = "sda2"
    volume_size = 16
  }

  ebs_block_device {
    device_name = "sda3"
    volume_size = 20
  }
}
```

- A. `aws_instance.example.ebs_block_device[*].volume_id`
- B. `aws_instance.example.ebs_block_device.volume_id`
- C. `aws_instance.example.ebs_block_device[sda2,sda3].volume_id`
- D. `aws_instance.example.ebs_block_device.*.volume_id`

Answer: A

Explanation:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/device_naming.html

NEW QUESTION 386

- (Exam Topic 4)

Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?

- A. Use terraform console command to have an interactive UI with full access to the underlying terraform state to run your interpolations , and debug at real-time.
- B. Add a breakpoint in your code, using the watch keyword , and output the value to console for temporary debugging.
- C. Use terraform zipmap function , it will be able to easily do the interpolations without complex code.
- D. Use terraform console command to have an interactive UI , but you can only use it with local state , and it does not work with remote state.

Answer: A

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 389

- (Exam Topic 4)

In a Terraform Cloud workspace linked to a version control repository, speculative plan runs start automatically when you merge or commit changes to version control.

- A. True
- B. False

Answer: B

NEW QUESTION 394

- (Exam Topic 4)

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. Tell the CIO that this is not possible . Resources created in CloudFormation , and AzureRM templates cannot be tracked using terraform.
- B. Use terraform import command to import each resource one by one .
- C. 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.
- D. 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.

Answer: B

NEW QUESTION 399

- (Exam Topic 4)

You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed?

- A. Terraform can still determine the correct order for destruction from the state even when you delete one or more items from the configuration.
- B. Those would be destroyed in the order in which they were written in the configuration file previously before you have deleted them from configuration file.
- C. The resource will be destructed in random order as you have already deleted them from configuration.
- D. You can not destroy resources by deleting them from configuration file and running plan and apply.

Answer: A

Explanation:

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

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

NEW QUESTION 401

- (Exam Topic 4)

By default, where does Terraform store its state file?

- A. Amazon S3 bucket
- B. shared directory
- C. remotely using Terraform Cloud
- D. current working directory

Answer: D

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 402

- (Exam Topic 4)

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

- A. name
- B. alias
- C. depends_on
- D. id

Answer: B

NEW QUESTION 407

- (Exam Topic 4)

A single terraform resource file that defines an aws_instance resource can simple be renamed to azurerm_virtual_machine in order to switch cloud providers

- A. True
- B. False

Answer: B

Explanation:

Providers usually require some configuration of their own to specify endpoint URLs, regions, authentication settings. Providers Initialization can be done by either explicitly via a provider block or by adding a resource from that provide <https://www.terraform.io/docs/configuration/providers.html>

NEW QUESTION 412

- (Exam Topic 4)

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: CD

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 415

- (Exam Topic 4)

If a Terraform creation-time provisioner fails, what will occur by default?

- A. The resource will not be affected, but the provisioner will need to be applied again
- B. The resource will be destroyed
- C. The resource will be marked as "tainted"
- D. Nothing, provisioners will not show errors in the command line

Answer: C

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 .

NEW QUESTION 417

- (Exam Topic 4)

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: D

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 418

- (Exam Topic 4)

terraform apply will fail if you have not run terraform plan first to update the plan output.

- A. True
- B. False

Answer: B

NEW QUESTION 421

- (Exam Topic 4)

Terraform configuration (including any module references) can contain only one Terraform provider type.

- A. True
- B. False

Answer: B

NEW QUESTION 422

- (Exam Topic 4)

When does Sentinel enforce policy logic during a Terraform Enterprise run?

- A. Before the plan phase
- B. During the plan phase
- C. Before the a apply phase
- D. After the apply phase

Answer: C

Explanation:

"Enforcing policy checks on runs - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is executed."

NEW QUESTION 426

- (Exam Topic 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. 5
- B. 50
- C. 10
- D. 20

Answer: C

NEW QUESTION 429

- (Exam Topic 4)

Named workspaces are not a suitable isolation mechanism for strong separation between staging and production?

- A. True
- B. False

Answer: A

Explanation:

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#when-to-use-multiple-workspaces>

NEW QUESTION 432

- (Exam Topic 4)

You're writing a Terraform configuration that needs to read input from a local file called id_rsa.pub. Which built-in Terraform function can you use to import the file's contents as a string?

- A. fileset("id_rsa.pub")
- B. filebase64("id_rsa.pub")
- C. templatefile("id_rsa.pub")
- D. file("id_rsa.pub")

Answer: D

Explanation:

<https://www.terraform.io/language/functions/file>

NEW QUESTION 433

- (Exam Topic 4)

Select all Operating Systems that Terraform is available for. (select five)

- A. Linux
- B. macOS
- C. Unix
- D. Solaris
- E. Windows
- F. FreeBSD

Answer: ABDEF

Explanation:

Terraform is available for macOS, FreeBSD, OpenBSD, Linux, Solaris, Windows <https://www.terraform.io/downloads.html>

NEW QUESTION 437

- (Exam Topic 4)

If a DevOps team adopts AWS Cloud Formation as their standardized method for provisioning public cloud resources, which of the following scenarios poses a challenge for this team?

- A. The team is asked to manage a new application stack built on AWS-native services
- B. The organization decides to expand into Azure and wishes to deploy new infrastructure using their existing codebase
- C. The team is asked to build a reusable code base that can deploy resources into any AWS region
- D. The DevOps team is tasked with automating a manual provisioning process

Answer: B

NEW QUESTION 439

- (Exam Topic 4)

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: AC

Explanation:

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 442

- (Exam Topic 4)

You've used Terraform to deploy a virtual machine and a database. You want to replace this virtual machine instance with an identical one without affecting the database. What is the best way to achieve this using Terraform?

- A. Use the Terraform taint command targeting the VMs then run Terraform plan and Terraform apply
- B. Delete the Terraform VM resources from your Terraform code then run Terraform plan and terraform apply
- C. Use the terraform apply command targeting the VM resources only
- D. Use the terraform state rm command to remove the VM from state file

Answer: A

Explanation:

<https://www.terraform.io/cli/state/taint>

NEW QUESTION 446

- (Exam Topic 4)

You cannot install third party plugins using terraform init.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/cli/commands/init>

For providers that are published in either the public Terraform Registry or in a third-party provider registry, terraform init will automatically find, download, and install the necessary provider plugins.

NEW QUESTION 448

- (Exam Topic 4)

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

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 452

- (Exam Topic 4)

Terraform plan updates your state file.

- A. True
- B. False

Answer: B

Explanation:

The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review. Source: <https://www.terraform.io/cli/commands/plan>

NEW QUESTION 456

- (Exam Topic 4)

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

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 458

- (Exam Topic 4)

Suppose terraformcode is taking up some values which are not defined inside the code files. In which of the following options issue might have occurred?

- A. Issue in main.tf file
- B. Issue in vars.tf file
- C. Issue in terraform.tfvars
- D. Issue in Environment Variables

Answer: D

NEW QUESTION 460

- (Exam Topic 4)

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. vpc_cidrs["us-east-1"]
- C. var.vpc_cidrs["us-east-1"]
- D. var.vpc_cidrs[0]

Answer: C

NEW QUESTION 462

- (Exam Topic 4)

What Terraform command can be used to inspect the current state file?

- A. terraform inspect
- B. terraform read

- C. terraform show
- D. terraform state

Answer: C

NEW QUESTION 463

- (Exam Topic 4)

Which are forbidden actions when the Terraform state file is locked? (Choose three.)

- A. terraform destroy
- B. terraform fmt
- C. terraform state list
- D. terraform apply
- E. terraform plan
- F. terraform validate

Answer: ADE

NEW QUESTION 464

- (Exam Topic 4)

```
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 dependenc
- B. Nothing else needs to be done from your end.
- C. Add explicit dependency using depends_on . This will ensure the correct order of 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.
- E. 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

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 468

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

TA-002-P Practice Exam Features:

- * TA-002-P Questions and Answers Updated Frequently
- * TA-002-P Practice Questions Verified by Expert Senior Certified Staff
- * TA-002-P Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * TA-002-P Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The TA-002-P Practice Test Here](#)