

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)

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 3

- (Exam Topic 1)

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

- A. split
- B. join
- C. slice
- D. chomp

Answer: C

Explanation:

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

NEW QUESTION 4

- (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 5

- (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 6

- (Exam Topic 1)

terraform init initializes a sample main.tf file in the current directory.

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 7

- (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 8

- (Exam Topic 1)

Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/downloads>

NEW QUESTION 9

- (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 10

- (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 10

- (Exam Topic 1)

You need to constrain the GitHub provider to version 2.1 or greater.

Which of the following should you put into the Terraform 0.12 configuration's provider block?

- A. version >= 2.1
- B. version ~> 2.1
- C. version = "<= 2.1"
- D. version = ">= 2.1"

Answer: D

Explanation:

version = ">= 1.2.0, < 2.0.0"

A version constraint is a string literal containing one or more conditions, which are separated by commas. Each condition consists of an operator and a version number.

Version numbers should be a series of numbers separated by periods (like 1.2.0), optionally with a suffix to indicate a beta release.

The following operators are valid:

= (or no operator): Allows only one exact version number. Cannot be combined with other conditions.

!=: Excludes an exact version number.

>, >=, <, <=: Comparisons against a specified version, allowing versions for which the comparison is true. "Greater-than" requests newer versions, and "less-than" requests older versions.

~>: Allows only the rightmost version component to increment. For example, to allow new patch releases within a specific minor release, use the full version number: ~> 1.0.4 will allow installation of 1.0.5 and 1.0.10 but not 1.1.0. This is usually called the pessimistic constraint operator.

<https://www.terraform.io/language/expressions/version-constraints>

NEW QUESTION 11

- (Exam Topic 1)

What is the workflow for deploying new infrastructure with Terraform?

- A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
- B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.
- C. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
- D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

Answer: D

Explanation:

Reference: <https://www.google.com/search?q=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&oq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&aqs=chrome..69i57j0j7&sourceid=chrome&ie=UTF-8>

NEW QUESTION 16

- (Exam Topic 1)

If a module declares a variable with a default, that variable must also be defined within the module.

- A. True
- B. False

Answer: B

NEW QUESTION 17

- (Exam Topic 1)

Terraform requires the Go runtime as a prerequisite for installation.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/plugin/sdkv2/guides/v1-upgrade-guide> and <https://www.terraform.io/plugin/sdkv2/guides/v2-upgrade-guide>

NEW QUESTION 21

- (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 22

- (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 25

- (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 28

- (Exam Topic 1)

What does the default "local" Terraform backend store?

- A. tfplan files
- B. Terraform binary
- C. Provider plugins
- D. State file

Answer: D

Explanation:

The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.

Reference: <https://www.terraform.io/docs/language/settings/backends/local.html>

NEW QUESTION 32

- (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 35

- (Exam Topic 1)

You have deployed a new webapp with a public IP address on a cloud provider. However, you did not create any outputs for your code.

What is the best method to quickly find the IP address of the resource you deployed?

- A. Run terraform output ip_address to view the result
- B. In a new folder, use the terraform_remote_state data source to load in the state file, then write an output for each resource that you find the state file
- C. Run terraform state list to find the name of the resource, then terraform state show to find the attributes including public IP address
- D. Run terraform destroy then terraform apply and look for the IP address in stdout

Answer: C

Explanation:

<https://www.terraform.io/cli/commands/state/show>

NEW QUESTION 39

- (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 44

- (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 46

- (Exam Topic 1)

If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?

- A. Run terraform refresh
- B. It will happen automatically
- C. Manually update the state file
- D. Run terraform import

Answer: A

Explanation:

[https://www.terraform.io/cli/commands/refresh#:~:text=The%20terraform%20refresh%20command%20reads%](https://www.terraform.io/cli/commands/refresh#:~:text=The%20terraform%20refresh%20command%20reads%20)

NEW QUESTION 47

- (Exam Topic 1)

Which backend does the Terraform CLI use by default?

- A. Terraform Cloud
- B. Consul
- C. Remote
- D. Local

Answer: D

Explanation:

"By default, Terraform implicitly uses a backend called local to store state as a local file on disk. Every other backend stores state in a remote service of some kind, which allows multiple people to access it. Accessing state in a remote service generally requires some kind of access credentials, since state data contains extremely sensitive information." <https://www.terraform.io/language/settings/backends>

NEW QUESTION 49

- (Exam Topic 2)

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

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

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

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

Answer: A

NEW QUESTION 51

- (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 53

- (Exam Topic 2)

Provisioners should only be used as a last resort.

- A. False
- B. True

Answer: B

Explanation:

Provisioners are a Last Resort

Terraform includes the concept of provisioners as a measure of pragmatism, knowing that there will always be certain behaviors that can't be directly represented in Terraform's declarative model.

However, they also add a considerable amount of complexity and uncertainty to Terraform usage. Firstly, Terraform cannot model the actions of provisioners as part of a plan because they can in principle take any action. Secondly, successful use of provisioners requires coordinating many more details than Terraform usage usually requires: direct network access to your servers, issuing Terraform credentials to log in, making sure that all of the necessary external software is installed, etc.

The following sections describe some situations which can be solved with provisioners in principle, but where better solutions are also available. We do not recommend using provisioners for any of the use-cases described in the following sections.

Even if your specific use-case is not described in the following sections, we still recommend attempting to solve it using other techniques first, and use provisioners only if there is no other option.

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

NEW QUESTION 57

- (Exam Topic 2)

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

- A. Local Backends
- B. This can be done using any of the local or remote backends
- C. Remote Backends
- D. Terraform Backends

Answer: C

Explanation:

The remote backend stores Terraform state and may be used to run operations in Terraform Cloud. When using full remote operations, operations like terraform plan or terraform apply can be executed in

Terraform Cloud's run environment, with log output streaming to the local terminal.

Remote plans and applies use variable values from the associated Terraform Cloud workspace. <https://www.terraform.io/docs/backends/types/remote.html>

NEW QUESTION 61

- (Exam Topic 2)

The terraform init command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing configuration or state.

- A. False
- B. True

Answer: B

Explanation:

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

NEW QUESTION 66

- (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 70

- (Exam Topic 2)

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

- A. True
- B. False

Answer: B

Explanation:

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

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

NEW QUESTION 74

- (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 77

- (Exam Topic 2)

terraform refresh will update 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. If the state is changed, this may cause changes to occur during the next plan or apply.

NEW QUESTION 78

- (Exam Topic 2)

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

- A. False
- B. True

Answer: A

Explanation:

You may see this QUESTION NO: in actual exam. Please remember : Terraform does not require GO language to be installed as a prerequisite and it does not require a Windows Server as well.

NEW QUESTION 79

- (Exam Topic 2)

You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?

- A. terraform unlock
- B. terraform force-unlock
- C. terraform state unlock
- D. None of the above

Answer: B

Explanation:

Command: force-unlock

Manually unlock the state for the defined configuration.

This will not modify your infrastructure. This command removes the lock on the state for the current configuration. The behavior of this lock is dependent on the backend being used. Local state files cannot be unlocked by another process.

<https://www.terraform.io/docs/commands/force-unlock.html> <https://www.terraform.io/docs/state/locking.html>

Terraform has a force-unlock command to manually unlock the state if unlocking failed.

If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed.

NEW QUESTION 80

- (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 83

- (Exam Topic 2)

Terraform must track metadata such as resource dependencies. Where is this data stored?

- A. workspace
- B. backend
- C. state file
- D. metadata store

Answer: C

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.

<https://www.terraform.io/docs/state/purpose.html#metadata>

NEW QUESTION 88

- (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 92

- (Exam Topic 3)

Which of the following is the right substitute for static values that can make Terraform configuration file more dynamic and reusable?

- A. Output value
- B. Input parameters
- C. Functions
- D. Modules

Answer: B

Explanation:

Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.

NEW QUESTION 95

- (Exam Topic 3)

Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the

modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.

- A. Create individual workspaces for each team , and ask them to share modules across workspaces.
- B. Implement a Private module registry in Terraform cloud , and ask teams to reference them.
- C. Upgrade to Terraform enterprise , since this is not possible in terraform cloud.
- D. Upload the modules in the terraform public module registry , and ask teams to reference them

Answer: B

Explanation:

Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster.

By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar.

Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise. Terraform Cloud's private module registry helps you share Terraform modules across your organization.

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

NEW QUESTION 98

- (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 101

- (Exam Topic 3)

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

- A. Local Backends.
- B. Terraform Backends.
- C. This can be done using any of the local or remote backends.
- D. Remote Backends.

Answer: D

Explanation:

When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.

Terraform Cloud can also be used with local operations, in which case only state is stored in the Terraform Cloud backend.

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

NEW QUESTION 105

- (Exam Topic 3)

Which of the below command will upgrade the provider version to the latest acceptable one?

- A. terraform plan upgrade
- B. terraform provider -upgrade
- C. terraform init -upgrade
- D. terraform init -update

Answer: C

Explanation:

To upgrade to the latest acceptable version of each provider, run terraform init -upgrade. This command also upgrades to the latest versions of all Terraform modules.

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

NEW QUESTION 108

- (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 109

- (Exam Topic 3)

Which of the below datatype is not supported by Terraform.

- A. Array
- B. List
- C. Object
- D. Map

Answer: A

NEW QUESTION 112

- (Exam Topic 3)

Command terraform refresh will update state file?

- A. False
- B. True

Answer: B

Explanation:

The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. This can be used to detect any drift from the last-known state, and to update the state file.

This does not modify infrastructure, but does modify the state file. If the state is changed, this may cause changes to occur during the next plan or apply. <https://www.terraform.io/docs/commands/refresh.html>

NEW QUESTION 117

- (Exam Topic 3)

You also have a defined the following environment variables in your shell: TF_itemNum =6, TF_VAR_itemNum =9. You also have a terraform.tfvars file with the following contents

itemNum = 7

When you run the following apply command, what is the value assigned to the itemNum variable? terraform apply -var itemNum =4

- A. 10
- B. 6
- C. 1
- D. 4
- E. 3

Answer: D

Explanation:

The -var and -var-file methods of assigning variables have the highest precedence. <https://www.terraform.io/docs/configuration/variables.html>

NEW QUESTION 118

- (Exam Topic 3)

Terraform-specific settings and behaviors are declared in which configuration block type?

- A. provider
- B. terraform
- C. resource
- D. data

Answer: B

Explanation:

The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.

```
Example terraform {  
  required_version = "> 0.12.0"  
}
```

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

NEW QUESTION 123

- (Exam Topic 3)

You have created a terraform script that uses a lot of new constructs that have been introduced in terraform v0.12. However, many developers who are cloning the script from your git repo, are using v0.11, and getting errors. What can be done from your end to solve this problem?

- A. Force developer to use v0.12 by using terraform setting 'required_version' and set it to >=0.12.
- B. Refactor the code to support both v0.11, and v0.12. It might be a difficult process, but there is no other way.
- C. Add a condition in front of each such specific construct, to check whether the running terraform version id v0.11 or v0.12, and ,work accordingly.
- D. Add comments in your code to tell developers to use v0.12 . If they use v0.11 , that should be their problem , which they need to figure out.

Answer: A

Explanation:

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

NEW QUESTION 125

- (Exam Topic 4)

Valarie has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code. Valarie wants to hide the output value in the CLI after terraform apply that's why she has used sensitive parameter.

```
* 1. output "db_password" {  
* 2. value = local.db_password  
* 3. sensitive = true  
* 4. }
```

Since sensitive is set to true, will the value associated with db password be available in plain-text in the state file for everyone to read?

- A. Yes
- B. No

Answer: A

Explanation:

Outputs can be marked as containing sensitive material by setting the sensitive attribute to true, like this: `output "sensitive" {
sensitive = true
value = VALUE
}`

When outputs are displayed on-screen following a terraform apply or terraform refresh, sensitive outputs are redacted, with <sensitive> displayed in place of their value.

Limitations of Sensitive Outputs

The values of sensitive outputs are still stored in the Terraform state, and available using the terraform output command, so cannot be relied on as a sole means of protecting values.

Sensitivity is not tracked internally, so if the output is interpolated in another module into a resource, the value will be displayed.

NEW QUESTION 129

- (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 133

- (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 136

- (Exam Topic 4)

Select the most accurate statement to describe the Terraform language from the following list.

- A. Terraform is an immutable, declarative, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally JSON.
- B. Terraform is a mutable, declarative, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.

C. Terraform is an immutable, procedural, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.

D. Terraform is a mutable, procedural, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally YAML.

Answer: A

Explanation:

Terraform is not a configuration management tool - <https://www.terraform.io/intro/vs/chefpuppet.html> Terraform is a declarative language -

<https://www.terraform.io/docs/configuration/index.html> Terraform supports a syntax that is JSON compatible <https://www.terraform.io/docs/configuration/syntax-json.html>

Terraform is primarily designed on immutable infrastructure principles - <https://www.hashicorp.com/resources/what-is-mutable-vs-immutable-infrastructure>

NEW QUESTION 140

- (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 144

- (Exam Topic 4)

How would you reference the attribute "name" of this fictitious resource in HCL?

```
resource "kubernetes_namespace" "example" {  
  name = "test"  
}
```

A. resource.kubrnetes_namespace>example.name

B. kubernetes_namespace.test.name

C. kubernetes_namespace.example.name

D. data kubernetes_namespace.name

E. None of the above

Answer: C

Explanation:

<https://www.terraform.io/language/expressions/references#references-to-resource-attributes>

NEW QUESTION 147

- (Exam Topic 4)

To check if all code in a Terraform configuration with multiple modules is properly formatted without making changes, what command should be run?

A. terraform fmt -check

B. terraform fmt -write=false

C. terraform fmt "list -recursive

D. terraform fmt -check -recursive

Answer: D

Explanation:

-check Check if the input is formatted. Exit status will be 0 if all input is properly formatted and non-zero otherwise.

-recursive Also process files in subdirectories. By default, only the given directory (or current directory) is processed.

NEW QUESTION 152

- (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 157

- (Exam Topic 4)

What are the benefits of using Infrastructure as Code? (select five)

- A. Infrastructure as Code is relatively simple to learn and write, regardless of a user's prior experience with developing code
- B. Infrastructure as Code provides configuration consistency and standardization among deployments
- C. Infrastructure as Code is easily repeatable, allowing the user to reuse code to deploy similar, yet different resources
- D. Infrastructure as Code gives the user the ability to recreate an application's infrastructure for disaster recovery scenarios
- E. Infrastructure as Code easily replaces development languages such as Go and .Net for application development
- F. Infrastructure as Code allows a user to turn a manual task into a simple, automated deployment (Correct)

Answer: ACDF

Explanation:

If you are new to infrastructure as code as a concept, it is the process of managing infrastructure in a file or files rather than manually configuring resources in a user interface.

A resource in this instance is any piece of infrastructure in a given environment, such as a virtual machine, security group, network interface, etc. At a high level, Terraform allows operators to use HCL to author files containing definitions of their desired resources on almost any provider (AWS, GCP, GitHub, Docker, etc) and automates the creation of those resources at the time of application.

NEW QUESTION 161

- (Exam Topic 4)

Which of the following can you do with terraform plan? Choose two correct answers.

- A. View the execution plan and check if the changes match your expectations
- B. Schedule Terraform to run at a planned time in the future
- C. Execute a plan in a different workspace
- D. Save a generated execution plan to apply later

Answer: AD

Explanation:

<https://learn.hashicorp.com/tutorials/terraform/plan>

NEW QUESTION 165

- (Exam Topic 4)

Why should secrets not be hard coded into Terraform code? Choose two correct answers

- A. All passwords should be rotated on a quarterly basis.
- B. The Terraform code is copied to the target resources to be applied locally and could expose secrets if a target resource is compromised.
- C. Terraform code is typically stored in version control, as well as copied to the systems from h it's run.Any of those may not have robust security mechanisms.
- D. It makes the code less reusable.

Answer: BC

NEW QUESTION 168

- (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 173

- (Exam Topic 4)

When using providers that require the retrieval of data, such as the HashiCorp Vault provider, in what phase does Terraform actually retrieve the data required?

- A. terraform delete
- B. terraform plan
- C. terraform init
- D. terraform apply

Answer: C

NEW QUESTION 177

- (Exam Topic 4)

Which parameters does terraform import require? Choose two correct answers.

- A. Provider
- B. Path
- C. Resource address
- D. Resource ID

Answer: CD

Explanation:

<https://www.terraform.io/cli/commands/import#usage>

NEW QUESTION 182

- (Exam Topic 4)

The Terraform CLI will print output values from a child module after running terraform apply.

- A. True
- B. False

Answer: A

NEW QUESTION 183

- (Exam Topic 4)

What does this code do?

```
terraform {
  required_providers {
    aws = "~> 3.0"
  }
}
```

- A. Requires any version of the AWS provider > = 3.0 and < 4.0
- B. Requires any version of the AWS provider > = 3.0
- C. Requires any version of the AWS provider after the 3.0 major release like 4.1
- D. Requires any version of the AWS provider > 3.0

Answer: A

Explanation:

<https://www.terraform.io/language/expressions/version-constraints#-3>

Allows only the rightmost version component to increment. For example, to allow new patch releases within a specific minor release, use the full version number:

~> 1.0.4 will allow installation of 1.0.5 and 1.0.10 but not 1.1.0

NEW QUESTION 187

- (Exam Topic 4)

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

- * 1. resource "aws_route53_record" "www"
- * 2. {
- * 3. zone_id = aws_route53_zone.primary.zone_id
- * 4. name = "www.example.com"

```
* 5. type = "A"  
* 6. ttl = "300"  
* 7. records = [module.web_server.instance_ip_address] 8. }
```

- A. The regular expression named module.web_server
- B. The output of a module named web_server
- C. By querying the AWS EC2 API to retrieve the IP address
- D. Value of the web_server parameter from the variables.tf file

Answer: B

Explanation:

In a parent module, outputs of child modules are available in expressions as module.<MODULE NAME>.<OUTPUT NAME>. For example, if a child module named web_server declared an output named instance_ip_address, you could access that value as module.web_server.instance_ip_address.

NEW QUESTION 189

- (Exam Topic 4)

Which of the following is true about Terraform's implementation of infrastructure as code? (Choose two.)

- A. It is only compatible with AWS infrastructure management
- B. You cannot reuse infrastructure configuration
- C. You can version your infrastructure configuration
- D. It requires manual configuration of infrastructure resources
- E. It allows you to automate infrastructure provisioning

Answer: CE

NEW QUESTION 192

- (Exam Topic 4)

Your configuration file has been locked accidentally. What of the following command would you use to unlock?

- A. terraform filename-unlock
- B. delete the file and create a new state file
- C. terraform force-unlock
- D. state.tf-unlock

Answer: C

NEW QUESTION 193

- (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 196

- (Exam Topic 4)

You are creating a Terraform configuration which needs to make use of multiple providers, one for AWS and one for Datadog.

Which of the following provider blocks would allow you to do this?

A)

```
provider {  
  "aws" {  
    profile = var.aws_profile  
    region  = var.aws_region  
  }  
  
  "datadog" {  
    api_key = var.datadog_api_key  
    app_key = var.datadog_app_key  
  }  
}
```

B)

```
provider "aws" {  
  profile = var.aws_profile  
  region  = var.aws_region  
}  
  
provider "datadog" {  
  api_key = var.datadog_api_key  
  app_key = var.datadog_app_key  
}
```

C)

```
terraform {  
  provider "aws" {  
    profile = var.aws_profile  
    region  = var.aws_region  
  }  
  
  provider "datadog" {  
    api_key = var.datadog_api_key  
    app_key = var.datadog_app_key  
  }  
}
```

- A. Option A
- B. Option B
- C. Option C

Answer: B

Explanation:

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

NEW QUESTION 198

- (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 199

- (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 204

- (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 207

- (Exam Topic 4)

You decide to move a Terraform state file to Amazon S3 from another location. You write the code below into a file called\

```
terraform {  
  backend "s3" {  
    bucket = "my-tf-bucket"  
    region = "us-east-1"  
  }  
}
```

You immediately run terraform apply but don't see any changes. Your state file didn't move. Which command will migrate your current state file to the new S3 remote backend?

A. terraform push

B. terraform init

C. terraform refresh

D. terraform state

Answer: B

NEW QUESTION 209

- (Exam Topic 4)

Which Terraform command will check and report errors within modules, attribute names, and value types to make sure they are syntactically valid and internally consistent?

A. terraform validate

B. terraform format

C. terraform fmt

D. terraform show

Answer: A

Explanation:

The terraform validate command validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs, etc.

Validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus primarily useful for general verification of reusable modules, including the correctness of attribute names and value types. It is safe to run this command automatically, for example as a post-save check in a text editor or as a test step for a re-usable module in a CI system.

NEW QUESTION 213

- (Exam Topic 4)

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

- A. False
- B. True

Answer: B

Explanation:

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

NEW QUESTION 217

- (Exam Topic 4)

What feature of Terraform Cloud and/or Terraform Enterprise can you publish and maintain a set of custom modules which can be used within your organization?

- A. Terraform registry
- B. custom VCS integration
- C. private module registry
- D. remote runs

Answer: C

NEW QUESTION 219

- (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 220

- (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 221

- (Exam Topic 4)

What does terraform import allow you to do?

- A. Import a new Terraform module
- B. Use a state file to import infrastructure to the cloud
- C. Import provisioned infrastructure to your state file
- D. Import an existing state file to a new Terraform workspace

Answer: C

NEW QUESTION 223

- (Exam Topic 4)

Module version is required to reference a module on the Terraform Module Registry.

- A. True
- B. False

Answer: B

NEW QUESTION 225

- (Exam Topic 4)

A terraform apply can not _____ infrastructure.

- A. import
- B. provision
- C. destroy
- D. change

Answer: A

NEW QUESTION 229

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

Explanation:

Splat Expressions with Maps The splat expression patterns shown above apply only to lists, sets, and tuples. To get a similar result with a map or object value you must use for expressions. Resources that use the for_each argument will appear in expressions as a map of objects, so you can't use splat expressions with those resources. For more information, see Referring to Resource Instances. https://www.terraform.io/language/meta-arguments/for_each#referring-to-instances
<https://www.terraform.io/language/expressions/references>

NEW QUESTION 233

- (Exam Topic 4)

The following is a snippet from a Terraform configuration file: Which, when validated, results in the following error:

Fill in the blank in the error message with the correct string from the list below.

- A. version
- B. multi
- C. label
- D. alias

Answer: D

Explanation:

<https://www.terraform.io/docs/configuration/providers.html#alias-multiple-providerinstances>

NEW QUESTION 234

- (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 237

- (Exam Topic 4)

Which of the following is not supported backend types in Terra form?

- A. consul
- B. gcs

- C. manta
- D. bitbucket

Answer: D

NEW QUESTION 241

- (Exam Topic 4)

terraform validate validate validates that your infrastructure matches the Terraform state file.

- A. True
- B. False

Answer: B

Explanation:

The terraform validate command validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs, etc. Validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus primarily useful for general verification of reusable modules, including correctness of attribute names and value types. Source: <https://www.terraform.io/cli/commands/validate>

NEW QUESTION 246

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 247

- (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 251

- (Exam Topic 4)

Which of the following is a meta-argument defined in the configuration files of Terraform?

- A. tfvar
- B. depends_on
- C. instance aws
- D. varl

Answer: B

NEW QUESTION 252

- (Exam Topic 4)

Choose the answer that correctly completes the sentence: _____ backends support state locking.

- A. All
- B. No
- C. Only local
- D. Some

Answer: D

NEW QUESTION 255

- (Exam Topic 4)

Why does this backend configuration not follow best practices?

```
terraform {  
  backend "s3" {  
    bucket     = "terraform-state-prod"  
    key        = "network/terraform.tfstate"  
    region     = "us-east-1"  
    access_key = "AKIAIOSFODNN7EXAMPLE"  
    secret_key = "wJalrXUtnFEMI/K7MDENG/bPxrRfiCYEXAMPLEKEY"  
  }  
  
  required_providers {  
    aws = {  
      source = "hashicorp/aws"  
      version = "~> 3.38"  
    }  
  }  
  
  required_version = ">= 0.15"  
}
```

- A. You should not store credentials in Terraform Configuration
- B. You should use the local enhanced storage backend whenever possible
- C. An alias meta-argument should be included in backend blocks whenever possible
- D. The backend configuration should contain multiple credentials so that more than one user can execute terraform plan and terraform apply

Answer: A

NEW QUESTION 256

- (Exam Topic 4)

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

- A. Terraform Modules
- B. Terraform Enterprise
- C. Terraform Workspaces
- D. Terraform Remote Backends

Answer: C

NEW QUESTION 261

- (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 262

- (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 266

- (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 271

- (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 273

- (Exam Topic 4)

How do you specify a module's version when publishing it to the public Terraform Module Registry?

- A. The module's configuration page on the Terraform Module Registry
- B. Terraform Module Registry does not support versioning modules
- C. The release tags in the associated repo Most Voted
- D. The module's Terraform code

Answer: C

Explanation:

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

NEW QUESTION 277

- (Exam Topic 4)

What does terraform refresh modify?

- A. Your cloud infrastructure
- B. Your Terraform plan
- C. Your state file
- D. Your Terraform configuration

Answer: C

NEW QUESTION 280

- (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 282

- (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 283

- (Exam Topic 4)

When do you need to explicitly execute terraform refresh?

- A. Before every terraform plan
- B. Before every terraform apply
- C. Before every terraform import
- D. None of the above

Answer: D

Explanation:

Wherever possible, avoid using terraform refresh explicitly and instead rely on Terraform's behavior of automatically refreshing existing objects as part of creating a normal plan. Source: <https://www.terraform.io/cli/commands/refresh>

NEW QUESTION 288

- (Exam Topic 4)

You need to write some Terraform code that adds 42 firewall rules to a security group as shown in the example.

```
resource "aws_security_group" "many_rules" {
  name = "many-rules"
  ingress {
    from_port = 443
    to_port = 443
    protocol = "tcp"
    cidr_blocks = "0.0.0.0/0"
  }
}
```

What can you use to avoid writing 42 different nested ingress config blocks by hand?

- A. A count loop
- B. A for block
- C. A for each block
- D. A 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. Reference: <https://www.terraform.io/language/expressions/dynamic-blocks>

NEW QUESTION 290

- (Exam Topic 4)

Which type of block fetches or computes information for use elsewhere in a Terraform configuration?

- A. provider
- B. resource
- C. local
- D. data

Answer: D

Explanation:

Data sources allow data to be fetched or computed for use elsewhere in Terraform configuration. Use of data sources allows a Terraform configuration to build on information defined outside of Terraform, or defined by another separate Terraform configuration.

NEW QUESTION 293

- (Exam Topic 4)

When you use a remote backend that needs authentication. HashrCorp recommends that you:

- A. Push your Terraform configuration to an encrypted git repository
- B. Write the authentication credentials in the Terraform configuration files
- C. Use partial configuration to load the authentication credentials outside of the Terraform code
- D. Keep the Terraform configuration files in a secret store

Answer: C

Explanation:

We recommend omitting the token from the configuration, and instead using terraform login or manually configuring credentials in the CLI config file. Reference: <https://www.terraform.io/language/settings/backends/remote>

NEW QUESTION 296

- (Exam Topic 4)

What advantage does an operations team that uses infrastructure as code have?

- A. The ability to delete infrastructure
- B. The ability to reuse best practice configurations and settings
- C. The ability to autoscale a group of servers
- D. The ability to update existing infrastructure

Answer: B

NEW QUESTION 300

- (Exam Topic 4)

Which of the following does terraform apply change after you approve the execution plan? Choose two correct answers.

- A. The execution plan
- B. Terraform code
- C. Cloud infrastructure
- D. State file
- E. The .terraform directory

Answer: CD

NEW QUESTION 304

- (Exam Topic 4)

What does state locking accomplish?

- A. Copies the state file from memory to disk
- B. Encrypts any credentials stored within the state file
- C. Blocks Terraform commands from modifying the state file
- D. Prevents accidental deletion of the state file

Answer: C

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

NEW QUESTION 309

- (Exam Topic 4)

What is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault?

- A. Vault provider
- B. API access using the AppRole auth method
- C. integration with a tool like Jenkins
- D. CLI access from the same machine running Terraform

Answer: A

NEW QUESTION 311

- (Exam Topic 4)

Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)

- A. Unix
- B. Red Hat
- C. CentOS
- D. Amazon Linux
- E. Ubuntu

Answer: BCDE

Explanation:

<https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements>

NEW QUESTION 313

- (Exam Topic 4)

You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration? 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:

Once you have authenticated to Terraform Cloud, you're ready to migrate your local state file to Terraform Cloud. To begin the migration, reinitialize. This causes

Terraform to recognize your cloud block configuration.

NEW QUESTION 318

- (Exam Topic 4)

True or False? By default, Terraform destroy will prompt for confirmation before proceeding.

- A. False
- B. True

Answer: B

NEW QUESTION 319

- (Exam Topic 4)

Your firm employs a version control system (for example, git) and has requested that you commit all terraform code to it. During the commit, you must be cautious with sensitive information. Which of the following files should be left out of the commit?

- A. main.tf
- B. variables.tf
- C. provisioner.tf
- D. terraform.tfstate

Answer: D

NEW QUESTION 323

- (Exam Topic 4)

You have just developed a new Terraform configuration for two virtual machines with a cloud provider. You would like to create the infrastructure for the first time. Which Terraform command should you run first?

- A. terraform apply
- B. terraform plan
- C. terraform show
- D. terraform init

Answer: D

NEW QUESTION 325

- (Exam Topic 4)

terraform apply is failing with the following error. What next step should you take to determine the root cause of the problem?

Error loading state: AccessDenied: Access Denied status code: 403, request id: 288766CE5CCA24A0, host id: FOOBAR

- A. Set TF_LOG=DEBUG
- B. Review syslog for Terraform error messages
- C. Run terraform login to reauthenticate with the provider
- D. Review /var/log/terraform.log for error messages

Answer: A

Explanation:

Terraform has detailed logs which can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr. You can set TF_LOG to one of the log levels (in order of decreasing verbosity) TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.

NEW QUESTION 326

- (Exam Topic 4)

In the following code snippet, the block type is identified by which string?

- A. "aws_instance"
- B. resource
- C. "db"
- D. instance_type

Answer: B

NEW QUESTION 330

- (Exam Topic 4)

Which of the below backends support state locking?

- A. S3
- B. consul
- C. azurerm
- D. artifactory

Answer: ABC

NEW QUESTION 333

- (Exam Topic 4)

What does terraform refresh modify?

- A. Your cloud infrastructure
- B. Your state file
- C. Your Terraform plan
- D. Your Terraform configuration

Answer: B

Explanation:

The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match. Source:
<https://www.terraform.io/cli/commands/refresh>

NEW QUESTION 338

- (Exam Topic 4)

You have been working in a Cloud provider account that is shared with other team members. You previously used Terraform to create a load balancer that is listening on port 80. After some application changes, you updated the Terraform code to change the port to 443.

You run terraform plan and see that the execution plan shows the port changing from 80 to 443 like you intended, and step away to grab some coffee.

In the meantime, another team member manually changes the load balancer port to 443 through the Cloud provider console before you get back to your desk.

What will happen when you terraform apply upon returning to your desk?

- A. Terraform will not make any changes to the Load Balancer and will update the state file to reflect any changes made.
- B. Terraform will change the port back to 80 in your code
- C. Terraform will change the load balancer port to 80, and then change it back to 443
- D. Terraform will fail with an error because the state file is no longer accurate

Answer: A

NEW QUESTION 340

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questions 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](#)