

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 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 5

- (Exam Topic 1)

terraform validate validates the syntax of Terraform files.

- A. True
- B. False

Answer: A

Explanation:

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

The terraform validate command validates the syntax and arguments of the Terraform configuration files. Reference:

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

NEW QUESTION 6

- (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 7

- (Exam Topic 1)

What information does the public Terraform Module Registry automatically expose about published modules?

- A. Required input variables
- B. Optional input variables and default values
- C. Outputs
- D. All of the above
- E. None of the above

Answer: D

Explanation:

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

"The registry extracts information about the module from the module's source. The module name, provider, documentation, inputs/outputs, and dependencies are all parsed and available via the UI or API, as well as the same information for any submodules or examples in the module's source repository."

NEW QUESTION 8

- (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 9

- (Exam Topic 1)

The terraform.tfstate file always matches your currently built infrastructure.

- A. True
- B. False

Answer: B

Explanation:

Reference: <https://www.terraform.io/docs/language/state/index.html>

NEW QUESTION 10

- (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 10

- (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 13

- (Exam Topic 1)

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

- A. The ability to share modules with public Terraform users and members of Terraform Enterprise Organizations
- B. The ability to tag modules by version or release
- C. The ability to restrict modules to members of Terraform Cloud or Enterprise organizations
- D. The ability to share modules publicly with any user of Terraform

Answer: C

Explanation:

Terraform Cloud's private registry works similarly to the public Terraform Registry and helps you share Terraform providers and Terraform modules across your organization. It includes support for versioning and a searchable list of available providers and modules.

NEW QUESTION 17

- (Exam Topic 1)

What is one disadvantage of using dynamic blocks in Terraform?

- A. They cannot be used to loop through a list of values
- B. Dynamic blocks can construct repeatable nested blocks
- C. They make configuration harder to read and understand
- D. Terraform will run more slowly

Answer: C

Explanation:

"Overuse of dynamic blocks can make configuration hard to read and maintain, so we recommend using them only when you need to hide details in order to build a clean user interface for a re-usable module. Always write nested blocks out literally where possible."

Reference: <https://github.com/hashicorp/terraform/issues/19291>

NEW QUESTION 18

- (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 22

- (Exam Topic 1)

You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead. What are the two things you must do to achieve this? (Choose two.)

- A. Provision new VMs using Terraform with the same VM names
- B. Use the terraform import command for the existing VMs
- C. Write Terraform configuration for the existing VMs
- D. Run the terraform import-gcp command

Answer: BC

Explanation:

You should create the equivalent configuration first, and then run import to load it on the state file.

NEW QUESTION 23

- (Exam Topic 1)

Which of these options is the most secure place to store secrets for connecting to a Terraform remote backend?

- A. Defined in Environment variables
- B. Inside the backend block within the Terraform configuration
- C. Defined in a connection configuration outside of Terraform
- D. None of above

Answer: A

Explanation:

<https://www.terraform.io/language/settings/backends/configuration#credentials-and-sensitive-data> Warning: We recommend using environment variables to supply credentials and other sensitive data. If you use -backend-config or hardcode these values directly in your configuration, Terraform will include these values in both the .terraform subdirectory and in plan files. This can leak sensitive credentials.

NEW QUESTION 25

- (Exam Topic 1)

Only the user that generated a plan may apply it.

- A. True
- B. False

Answer: B

NEW QUESTION 30

- (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 33

- (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 37

- (Exam Topic 1)

FILL BLANK

Which flag would you add to terraform plan to save the execution plan to a file?

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

"You can use the optional -out=FILE option to save the generated plan to a file on disk, which you can later execute by passing the file to terraform apply as an extra argument. This two-step workflow is primarily intended for when running Terraform in automation. If you run terraform plan without the -out=FILE option then it will create a speculative plan, which is a description of the effect of the plan but without any intent to actually apply it." <https://www.terraform.io/cli/commands/plan>

NEW QUESTION 42

- (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 45

- (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 49

- (Exam Topic 1)

Terraform variables and outputs that set the "description" argument will store that description in the state file.

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 50

- (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 54

- (Exam Topic 1)

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

```
mainresource "google_compute_instance" "main" {  
    name = "test"  
}
```

- A. compute_instance
- B. main
- C. google
- D. test

Answer: B

NEW QUESTION 58

- (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 59

- (Exam Topic 1)

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

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

- A. Run terraform 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 62

- (Exam Topic 1)

A Terraform provider is not responsible for:

- A. Understanding API interactions with some service
- B. Provisioning infrastructure in multiple clouds
- C. Exposing resources and data sources based on an API
- D. Managing actions to take based on resource differences

Answer: B

Explanation:

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

NEW QUESTION 64

- (Exam Topic 1)

What is terraform refresh intended to detect?

- A. Terraform configuration code changes
- B. Empty state files
- C. State file drift
- D. Corrupt state files

Answer: C

Explanation:

"The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match. Warning: This command is deprecated, because its default behavior is unsafe if you have misconfigured credentials for any of your providers. See below for more information and

recommended alternatives." <https://www.terraform.io/cli/commands/refresh>

NEW QUESTION 65

- (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 68

- (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 70

- (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 74

- (Exam Topic 1)

All standard backend types support state storage, locking, and remote operations like plan. apply and destroy.

- A. True
- B. False

Answer: B

Explanation:

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

"Some of these backends act like plain remote disks for state files, while others support locking the state while operations are being performed. This helps prevent conflicts and inconsistencies. The built-in backends listed are the only backends. You cannot load additional backends as plugins."

NEW QUESTION 75

- (Exam Topic 1)

Where in your Terraform configuration do you specify a state backend?

- A. The terraform block
- B. The resource block
- C. The provider block
- D. The datasource block

Answer: A

Explanation:

Backends are configured with a nested backend block within the top-level terraform block. Reference:

<https://www.terraform.io/docs/language/settings/backends/configuration.html> <https://www.terraform.io/language/settings/backends/configuration#using-a-backend-block>

NEW QUESTION 77

- (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 80

- (Exam Topic 1)

How is terraform import run?

- A. As a part of terraform init
- B. As a part of terraform plan
- C. As a part of terraform refresh
- D. By an explicit call
- E. All of the above

Answer: D

Explanation:

"The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration. Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped. While this may seem tedious, it still gives Terraform users an avenue for importing existing resources."

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

NEW QUESTION 81

- (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 82

- (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 83

- (Exam Topic 2)

John wants to use two different regions to deploy two different EC2 instances. He has specified two provider blocks in his providers.tf file.

```
provider "aws" { region = "us-east-1" } provider "aws" { region = "us-west-2" }
```

When he run terraform plan he encountered an error. How to fix this?

- A. Use another provider version
- B. Use alias for region = "us-west-2"
- C. Use default keyword with region = "us-east-1"
- D. It can not be fixed

Answer: B

NEW QUESTION 88

- (Exam Topic 2)

If you enable TF_LOG = DEBUG, the log will be stored in syslog.log file in the current directory.

- A. False
- B. True

Answer: A

Explanation:

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

NEW QUESTION 93

- (Exam Topic 2)

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

- A. Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using Policy-As-Code.
- B. Implement a review process where every code will be reviewed before merging to the master branch.
- C. Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and writePolicy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.
- D. Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.

Answer: C

Explanation:

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

NEW QUESTION 95

- (Exam Topic 2)

You want to use terraform import to start managing infrastructure that was not originally provisioned through infrastructure as code. Before you can import the resource's current state, what must you do in order to prepare to manage these resources using Terraform?

- A. Run terraform refresh to ensure that the state file has the latest information for existing resources.
- B. Update the configuration file to include the new resources.
- C. Shut down or stop using the resources being imported so no changes are inadvertently missed.
- D. Modify the Terraform state file to add the new resources.

Answer: B

Explanation:

The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration.

Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.

The terraform import command is used to import existing infrastructure.

To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.

Example:

```
resource "aws_instance" "import_example" {  
# ...instance configuration...  
}
```

Now terraform import can be run to attach an existing instance to this resource configuration.

```
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
```

```
aws_instance.import_example: Import complete!
```

```
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
```

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 98

- (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 102

- (Exam Topic 2)

The current implementation of Terraform import can only import resources into the state. It does not generate configuration.

- A. False
- B. True

Answer: B

Explanation:

The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration.

Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.

While this may seem tedious, it still gives Terraform users an avenue for importing existing resources. <https://www.terraform.io/docs/import/index.html#currently-state-only>

NEW QUESTION 107

- (Exam Topic 2)

Terraform import command can import resources into modules as well directly into the root of your state.

A. True

B. False

Answer: A

Explanation:

Import will find the existing resource from ID and import it into your Terraform state at the given ADDRESS. ADDRESS must be a valid resource address. Because any resource address is valid, the import command can import resources into modules as well directly into the root of your state.

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. For 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.

As a result of the above command, the resource is recorded in the state file. We can now run terraform plan to see how the configuration compares to the imported resource, and make any adjustments to the configuration to align with the current (or desired) state of the imported object.

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

NEW QUESTION 110

- (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 112

- (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 113

- (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 114

- (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 117

- (Exam Topic 2)

You have declared a variable name my_var in terraform configuration without a value associated with it. variable my_var {}

After running terraform plan it will show an error as variable is not defined.

A. True

B. False

Answer: B

Explanation:

Input variables are usually defined by stating a name, type and a default value. However, the type and default values are not strictly necessary. Terraform can deduct the type of the variable from the default or input value.

Variables can be predetermined in a file or included in the command-line options. As such, the simplest variable is just a name while the type and value are selected based on the input.

```
variable "variable_name" {}
```

```
terraform apply -var variable_name="value"
```

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

String

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

```
variable "template" { type = string
```

```
default = "01000000-0000-4000-8000-000030080200"
```

```
}
```

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

```
storage = var.template List
```

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

Here is an example of a list variable definition.

```
variable "users" { type = list
```

```
default = ["root", "user1", "user2"]
```

```
}
```

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

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

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

```
variable "plans" { type = map default = {
```

```
"5USD" = "1xCPU-1GB" "10USD" = "1xCPU-2GB" "20USD" = "2xCPU-4GB"
```

```
}
```

```
}
```

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

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

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

```
variable "storage_sizes" { type = map
```

```
default = {
```

```
"1xCPU-1GB" = "25"
```

```
"1xCPU-2GB" = "50"
```

```
"2xCPU-4GB" = "80"
```

```
}
```

```
}
```

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

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

Boolean

The last of the available variable type is boolean. They give the option to employ simple true or false values. For example, you might wish to have a variable that decides when to generate the root user password on a new deployment.

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

```
}
```

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

```
create_password = var.set_password
```

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

variable.
terraform apply -var set_password="true"

NEW QUESTION 122

- (Exam Topic 2)

What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

- A. Local backends
- B. Providers
- C. Remote backends
- D. Workspaces

Answer: D

Explanation:

Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
<https://www.terraform.io/docs/state/workspaces.html>

NEW QUESTION 127

- (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 129

- (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 133

- (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 136

- (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 141

- (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 143

- (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 144

- (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 148

- (Exam Topic 3)

Once a resource is marked as tainted, the next plan will show that the resource will be _____ and _____ and the next apply will implement this change.

- A. recreated and tainted
- B. destroyed and not recreated
- C. tainted and not destroyed
- D. destroyed and recreated

Answer: D

NEW QUESTION 152

- (Exam Topic 3)

Which of the following allows Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code?

- A. Sentinel
- B. Module registry
- C. Functions
- D. Workspaces

Answer: A

Explanation:

Sentinel is a language and framework for policy built to be embedded in existing software to enable fine-grained, logic-based policy decisions. A policy describes under what circumstances certain behaviors are allowed. Sentinel is an enterprise-only feature.
https://www.youtube.com/watch?v=Vy8s7AAvU6g&feature=emb_title

NEW QUESTION 153

- (Exam Topic 3)

Dawn has created the below child module. Without changing the module, can she override the instance_type from t2.micro to t2.large from her code while calling this module?

```
* 1. resource "aws_instance" "myec2"
* 2. {
* 3. ami = "ami-082b5a644766e0e6f"
* 4. instance_type = "t2.micro
* 5. }
```

- A. YES
- B. No

Answer: B

Explanation:

As the instance_type is hard-coded in source module, you will not be able to change its value from destination module. Instead of hard-coding you should use variable with default values.

NEW QUESTION 156

- (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 159

- (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 162

- (Exam Topic 3)

If you delete a remote backend from the configuration, will you need to rebuild your state files locally?

- A. False
- B. True

Answer: A

Explanation:

You can change your backend configuration at any time. You can change both the configuration itself as well as the type of backend (for example from "consul" to "s3").

Terraform will automatically detect any changes in your configuration and request a reinitialization. As part of the reinitialization process, Terraform will ask if you'd like to migrate your existing state to the new configuration. This allows you to easily switch from one backend to another.

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

NEW QUESTION 166

- (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 169

- (Exam Topic 3)

Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.

- A. terraform state
- B. terraform apply
- C. terraform plan
- D. terraform validate
- E. terraform output

Answer: BC

NEW QUESTION 170

- (Exam Topic 3)

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

- A. Use Sentinel Policies.
- B. Use S3 bucket policy.
- C. Create a script that checks the encryption parameter is enabled on every git commit.
- D. Shared a SOP to engineers to mandate encryption feature on S3.

Answer: A

Explanation:

Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.

Using Sentinel with Terraform Cloud involves:

* Defining the policies - Policies are defined using the policy language with imports for parsing the Terraform plan, state and configuration.

* Managing policies for organizations - Users with permission to manage policies can add policies to their organization by configuring VCS integration or uploading policy sets through the API. They also define which workspaces the policy sets are checked against during runs. (More about permissions.)

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

* Mocking Sentinel Terraform data - Terraform Cloud provides the ability to generate mock data for any run within a workspace. This data can be used with the Sentinel CLI to test policies before deployment.

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

NEW QUESTION 171

- (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 173

- (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 174

- (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 176

- (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 178

- (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 180

- (Exam Topic 3)

Hanah is writing a terraform configuration with nested modules, there are multiple places where she has to use the same conditional expression but she wants to avoid repeating the same values or expressions multiple times in the configuration,. What is a better approach to dealing with this?

- A. Expressions

- B. Local Values
- C. Variables
- D. Functions

Answer: B

Explanation:

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

NEW QUESTION 185

- (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 189

- (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 192

- (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 is 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 193

- (Exam Topic 3)

Which of the following are string functions? Select three

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

Answer: CDE

Explanation:

tonumber and tostring are Type Conversion functions <https://www.terraform.io/docs/configuration/functions.html>

NEW QUESTION 195

- (Exam Topic 3)

Mary has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code:

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

Mary wants to hide the output value in the CLI after terraform apply? What is the best way?

- A. Use secure parameter
- B. Use sensitive parameter
- C. Use cryptographic hash

D. Encrypt the value using encrypt() function

Answer: B

NEW QUESTION 199

- (Exam Topic 3)

During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

- A. The resource will be planned for destruction and recreation upon the next terraform apply
- B. Terraform will retry to provision again.
- C. The failure of provisioner will be ignored and it will not cause a failure to terraform apply
- D. The resource will be automatically destroyed.

Answer: A

Explanation:

If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply. Terraform does this because a failed provisioner can leave a resource in a semi-configured state. Because Terraform cannot reason about what the provisioner does, the only way to ensure proper creation of a resource is to recreate it. This is tainting.

You can change this behavior by setting the on_failure attribute, which is covered in detail below. <https://www.terraform.io/docs/provisioners/index.html#creation-time-provisioners> <https://www.terraform.io/docs/provisioners/index.html#destroy-time-provisioners> <https://www.terraform.io/docs/provisioners/index.html#failure-behavior>

NEW QUESTION 202

- (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 203

- (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 204

- (Exam Topic 4)

terraform init retrieves the source code for all referenced modules

- A. True
- B. False

Answer: A

Explanation:

Terraform installs providers, initialises source code & modules etc at this stage

NEW QUESTION 206

- (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 208

- (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 212

- (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 215

- (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 216

- (Exam Topic 4)

A user runs terraform init on their RHEL based server and per the output, two provider plugins are downloaded: \$ terraform init

Initializing the backend... Initializing provider plugins...

- Checking for available provider plugins...

- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...

- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...

:

Terraform has been successfully initialized! Where are these plugins downloaded to?

- A. The .terraform.plugins directory in the directory terraform init was executed in.
- B. The .terraform/plugins directory in the directory terraform init was executed in.
- C. /etc/terraform/plugins
- D. The .terraform.d directory in the directory terraform init was executed in.

Answer: B

NEW QUESTION 219

- (Exam Topic 4)

Which of the following actions are performed during a terraform init?

- A. Initializes downloaded and/or installed providers
- B. Initializes the backend configuration
- C. Provisions the declared resources in your configuration
- D. Download the declared providers which are supported by HashiCorp

Answer: ABD

Explanation:

The terraform init command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times. This 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. terraform init command does

- * Copy a Source Module
- * Backend Initialization
- * Child Module Installation
- * Plugin Installation <https://www.terraform.io/docs/commands/init.html>

NEW QUESTION 222

- (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 224

- (Exam Topic 4)

Which one is the right way to import a local module names consul?

- A. module "consul" { source = "consul"}
- B. module "consul" { source = "./consul"}
- C. module "consul" { source = "../consul"}
- D. module "consul" { source = "module/consul"}

Answer: BC

Explanation:

A local path must begin with either ./ or ../ to indicate that a local path is intended, to distinguish from a module registry address.

```
module "consul" {  
  source = "./consul"  
}
```

NEW QUESTION 226

- (Exam Topic 4)

Open source Terraform can only import publicly-accessible and open-source modules.

- A. True
- B. False

Answer: B

Explanation:

Terraform can load modules from a public or private registry. This makes it possible to publish modules for others to use, and to use modules that others have published. Also, members of your organization might produce modules specifically crafted for your own infrastructure needs. Terraform Cloud and Terraform Enterprise both include a private module registry for sharing modules internally within your organization. Source: <https://www.terraform.io/language/modules>

NEW QUESTION 227

- (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 228

- (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 233

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 236

- (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 239

- (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 243

- (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 247

- (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 248

- (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 253

- (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 257

- (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 261

- (Exam Topic 4)

Which task does terraform init not perform?

- A. Sources any modules and copies the configuration locally
- B. Validates all required variables are present
- C. Connects to the backend
- D. Sources all providers present in the configuration and ensures they are downloaded and available locally

Answer: B

NEW QUESTION 264

- (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 266

- (Exam Topic 4)

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

```
data "vsphere_datacenter" "dc" {}

resource "vsphere_folder" "parent" {
    path = "Production"
    type = "vm"
    datacenter id = _____
}
```

- A. vsphere_datacenter.dc.id
- B. data.vsphere_datacenter.dc
- C. data.dc.id
- D. data.vsphere_datacenter.dc.id

Answer: D

NEW QUESTION 268

- (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 269

- (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 272

- (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 275

- (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 278

- (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 281

- (Exam Topic 4)

When should Terraform configuration files be written when running terraform import on existing infrastructure?

- A. Infrastructure can be imported without corresponding Terraform code
- B. Terraform will generate the corresponding configuration files for you
- C. You should write Terraform configuration files after the next terraform import is executed
- D. Terraform configuration should be written before terraform import is executed

Answer: D

Explanation:

The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration.

Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.

Source: <https://www.terraform.io/cli/import>

NEW QUESTION 282

- (Exam Topic 4)

terraform destroy is the only way to remove infrastructure.

- A. True
- B. False

Answer: B

NEW QUESTION 286

- (Exam Topic 4)

Resources in terraform can have same identifiers(Resource type + Block name).

- A. True
- B. False

Answer: B

NEW QUESTION 288

- (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 293

- (Exam Topic 4)

True or False: Workspaces provide identical 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/workspaces/index.html> <https://www.terraform.io/docs/state/workspaces.html>

NEW QUESTION 294

- (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 297

- (Exam Topic 4)

You have created a custom variable definition file my_vars.tfvars. How will you use it for provisioning infrastructure?

- A. terraform apply -var-state-file="my_vars.tfvars"
- B. terraform apply var-file="my_vars.tfvars"
- C. terraform plan -var-file="my_vars.tfvar"
- D. terraform apply -var-file="my_vars.tfvars"

Answer: D

Explanation:

To set lots of variables, it is more convenient to specify their values in a variable definitions file (with a filename ending in either .tfvars or .tfvars.json) and then specify that file on the command line with -var-file:

terraform apply -var-file="my_vars.tfvars" <https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files>

NEW QUESTION 302

- (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 307

- (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 310

- (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 311

- (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 315

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 319

- (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 322

- (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 324

- (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 326

- (Exam Topic 4)

A terraform apply can not _____ infrastructure.

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

Answer: A

NEW QUESTION 329

- (Exam Topic 4)

Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?

- A. False
- B. True

Answer: B

Explanation:

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

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

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

NEW QUESTION 331

- (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 333

- (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 335

- (Exam Topic 4)

In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to use what?

- A. Input Parameters
- B. Module
- C. Regular Expressions
- D. Output Value

Answer: A

Explanation:

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

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

NEW QUESTION 337

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 340

- (Exam Topic 4)

From the answers below, select the advantages of using Infrastructure as Code.

- A. Provide a codified workflow to develop customer-facing applications.
- B. Safely test modifications using a "dry run" before applying any actual changes.
- C. Easily integrate with application workflows (GitLab Actions, Azure DevOps, CI/CD tools).
- D. Easily change and update existing infrastructure.
- E. Provide reusable modules for easy sharing and collaboration.

Answer: BCDE

Explanation:

Infrastructure as Code is not used to develop applications, but it can be used to help deploy or provision those applications to a public cloud provider or on-premises infrastructure.

All of the others are benefits to using Infrastructure as Code over the traditional way of managing infrastructure, regardless if it's public cloud or on-premises.

NEW QUESTION 341

- (Exam Topic 4)

Jack is a newbie to Terraform and wants to enable detailed logging to find all the details. Which environment variable does he need to set?

- A. TF_help
- B. TF_LOG
- C. TF_Debug
- D. TF_var_log

Answer: B

NEW QUESTION 343

- (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 348

- (Exam Topic 4)

Select the feature below that best completes the sentence:

The following list represents the different types of _____ available in Terraform.

- * 1. max
- * 2. min
- * 3. join
- * 4. replace
- * 5. list
- * 6. length
- * 7. range

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

Answer: D

Explanation:

The Terraform language includes a number of built-in functions that you can call from within expressions to transform and combine values. The Terraform language does not support user-defined functions, and only the functions built into the language are available for use.
<https://www.terraform.io/docs/configuration/functions.html>

NEW QUESTION 349

- (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 354

- (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 359

- (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 363

- (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 367

- (Exam Topic 4)

Which are examples of infrastructure as code? (Choose two.)

- A. Cloned virtual machine images
- B. Change management database records
- C. Versioned configuration files
- D. Docker files

Answer: CD

NEW QUESTION 370

- (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 375

- (Exam Topic 4)

During a terraform plan, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

- A. Terraform attempts to provision the resource up to three times before exiting with an error
- B. the terraform plan is rolled back and all provisioned resources are removed
- C. it is automatically deleted
- D. the resource is marked as tainted

Answer: D

Explanation:

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

NEW QUESTION 377

- (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 380

- (Exam Topic 4)

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:

depends_on

NEW QUESTION 383

- (Exam Topic 4)

Which of the following is true about terraform apply? (Choose two.)

- A. It only operates on infrastructure defined in the current working directory or workspace
- B. You must pass the output of a terraform plan command to it
- C. Depending on provider specification, Terraform may need to destroy and recreate your infrastructure resources
- D. By default, it does not refresh your state file to reflect current infrastructure configuration
- E. You cannot target specific resources for the operation

Answer: AC

Explanation:

<https://www.terraform.io/cli/run>

NEW QUESTION 388

- (Exam Topic 4)

When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)

- A. directly querying HashiCorp Vault for the secrets
- B. command-line key/value pairs
- C. use the -backend-config=PATH to specify a separate config file
- D. interactively on the command line

Answer: BCD

Explanation:

You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.

With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments: <https://www.terraform.io/docs/backends/init.html#backend-initialization>

NEW QUESTION 390

- (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 395

- (Exam Topic 4)

Which of the following is the safest way to inject sensitive values into a Terraform Cloud workspace?

- A. Write the value to a file and specify the file with the -var-file flag
- B. Set a value for the variable in the UI and check the "Sensitive" check box
- C. Edit the state file directly just before running terraform apply
- D. Set the variable value on the command line with the -var flag

Answer: B

Explanation:

-var and -var-file overwrite workspace-specific and variable set variables that have the same key. From the workspace, variable can be added and checked off as being sensitive. Reference: <https://www.terraform.io/cloud-docs/workspaces/variables/managing-variables#loading-variables-from-files>
<https://www.terraform.io/cloud-docs/workspaces/variables>

NEW QUESTION 400

- (Exam Topic 4)

terraform validate reports HCL syntax errors.

- A. True
- B. False

Answer: A

NEW QUESTION 403

- (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 406

- (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 409

- (Exam Topic 4)

A junior admin accidentally deleted some of your cloud instances. What does Terraform do when you run terraform apply?

- A. Build a completely brand new set of infrastructure
- B. Tear down the entire workspace infrastructure and rebuild it
- C. Rebuild only the instances that were deleted Most Voted
- D. Stop and generate an error message about the missing instances

Answer: C

NEW QUESTION 412

- (Exam Topic 4)

Which of the following is not an advantage of using infrastructure as code operations?

- A. Self-service infrastructure deployment
- B. Troubleshoot via a Linux diff command
- C. Public cloud console configuration workflows
- D. Modify a count parameter to scale resources
- E. API driven workflows

Answer: B

Explanation:

terraform is used to deploy the infrastructure, not to troubleshoot it

NEW QUESTION 416

- (Exam Topic 4)

You have a Terraform configuration that defines a single virtual machine with no references to it. You have run terraform apply to create the resource, and then removed the resource definition from your Terraform configuration file.

What will happen when you run terraform apply in the working directory again?

- A. Nothing
- B. Terraform will destroy the virtual machine
- C. Terraform will error
- D. Terraform will remove the virtual machine from the state file, but the resource will still exist

Answer: B

Explanation:

If you remove the resource from your config file and the resource is in your state file, terraform will apply the configuration in the config file - which is to delete the resource

NEW QUESTION 418

- (Exam Topic 4)

Terraform Cloud is available only as a paid offering from HashiCorp.

- A. True
- B. False

Answer: B

Explanation:

Many of Terraform Cloud features are free for small teams, including remote state storage, remote runs, and VCS connections.

"Terraform Cloud is a commercial SaaS product developed by HashiCorp. Many of its features are free for small teams, including remote state storage, remote

runs, and VCS connections. We also offer paid plans for larger teams that include additional collaboration and governance features."

NEW QUESTION 420

- (Exam Topic 4)

What does Terraform use providers for? (Choose three.)

- A. Provision resources for on-premises infrastructure services
- B. Simplify API interactions
- C. Provision resources for public cloud infrastructure services
- D. Enforce security and compliance policies
- E. Group a collection of Terraform configuration files that map to a single state file

Answer: ABC

NEW QUESTION 423

- (Exam Topic 4)

What are some of the features of Terraform state? (select three)

- A. inspection of cloud resources
- B. determining the correct order to destroy resources
- C. mapping configuration to real-world resources
- D. increased performance

Answer: CD

NEW QUESTION 425

- (Exam Topic 4)

Your team lead does not trust the junior terraform engineers who now have access to the git repo . So , he wants you to have some sort of a checking layer , whereby , you can ensure that the juniors will not create any non-compliant resources that might lead to a security audit failure in future. What can you do to efficiently enforce this?

- A. Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.
- B. Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and writePolicy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.
- C. Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using Policy-As-Code.
- D. Create a git master branch , and implement PR . Every change needs to be reviewed by you , before being merged to the master branch.

Answer: B

Explanation:

Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.

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

NEW QUESTION 430

- (Exam Topic 4)

You are using a networking module in your Terraform configuration with the name label my_network. In your main configuration you have the following code:

```
output: "net_id" {  
  value = module.my_network.vnet_id  
}
```

When you run terraform validate, you get the following error:

```
Error: Reference to undeclared output value  
  
on main.tf line 12, in output "net_id":  
12:   value = module.my_network.vnet_id
```

What must you do to successfully retrieve this value from your networking module?

- A. Define the attribute vnet_id as a variable in the networking module
- B. Change the referenced value to module.my_network.outputs.vnet_id
- C. Define the attribute vnet_id as an output in the networking module
- D. Change the referenced value to my_network.outputs.vnet_id

Answer: C

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_addr, you could access that value as module.web_server.instance_ip_addr.

NEW QUESTION 434

- (Exam Topic 4)

In the below configuration, how would you reference the module output vpc_id?

```
module "vpc" {  
  source = "terraform-and-modules/vpc/aws"  
  cidr = "10.0.0.0/16"  
  name = "test-vpc"  
}
```

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:

<https://cloudcasts.io/course/terraform/community-vpc-module>

NEW QUESTION 437

- (Exam Topic 4)

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

- A. False
- B. True

Answer: B

NEW QUESTION 442

- (Exam Topic 4)

As a member of the operations team, you need to run a script on a virtual machine created by Terraform. Which provisioner is best to use in your Terraform code?

- A. local-exec
- B. file
- C. null-exec
- D. remote-exec

Answer: D

Explanation:

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

NEW QUESTION 446

- (Exam Topic 4)

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

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

Answer: E

Explanation:

Terraform CLI defines the following OPTIONAL arguments for variable declarations: default - A default value which then makes the variable optional. type - This argument specifies what value types are accepted for the variable. description - This specifies the input variable's documentation. validation - A block to define validation rules, usually in addition to type constraints. sensitive - Limits Terraform UI output when the variable is used in configuration. nullable - Specify if the variable can be null within the module. <https://www.terraform.io/language/values/variables#arguments>

NEW QUESTION 451

- (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 452

- (Exam Topic 4)

Which of the following is not a way to trigger terraform destroy ?

- A. Passing ---destroy at the end of apian request
- B. Running terraform destroy from the correct directory and then typing "yes" when prompted in the CLI
- C. Using the destroy command with auto approve
- D. Delete the state file and run terraform apply

Answer: A

NEW QUESTION 454

- (Exam Topic 4)

Which of the following connection types are supported by the remote-exec provisioner? (select two)

- A. WinRM
- B. UDP
- C. SMB
- D. RDP
- E. ssh

Answer: AE

Explanation:

The remote-exec provisioner invokes a script on a remote resource after it is created. The remote-exec provisioner supports both ssh and winrm type connections. remote-exec connection types

* ssh on Linux

* winrm on Windows <https://www.terraform.io/docs/provisioners/remote-exec.html>

NEW QUESTION 456

- (Exam Topic 4)

What is a key benefit of the Terraform state file?

- A. A state file represents a source of truth for resources provisioned with a public cloud console
- B. A state file represents a source of truth for resources provisioned with Terraform
- C. A state file represents the desired state expressed by the Terraform code files
- D A state file can be used to schedule recurring infrastructure tasks

Answer: C

NEW QUESTION 460

.....

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](#)