

## TA-002-P Dumps

### HashiCorp Certified: Terraform Associate

<https://www.certleader.com/TA-002-P-dumps.html>



**NEW QUESTION 1**

- (Exam Topic 1)

When you initialize Terraform, where does it cache modules from the public Terraform Module Registry?

- A. On disk in the /tmp directory
- B. In memory
- C. On disk in the .terraform sub-directory
- D. They are not cached

**Answer:** C

**Explanation:**

"A hidden .terraform directory, which Terraform uses to manage cached provider plugins and modules, record which workspace is currently active, and record the last known backend configuration in case it needs to migrate state on the next run. This directory is automatically managed by Terraform, and is created during initialization." <https://www.terraform.io/cli/init>

**NEW QUESTION 2**

- (Exam Topic 1)

Which option can not be used to keep secrets out of Terraform configuration files?

- A. A Terraform provider
- B. Environment variables
- C. A -var flag
- D. secure string

**Answer:** A

**Explanation:**

Reference: <https://secrethub.io/blog/secret-management-for-terraform/>

**NEW QUESTION 3**

- (Exam Topic 1)

You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you delete the VM using the cloud provider console, and run terraform apply again without changing any Terraform code?

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

**Answer:** D

**NEW QUESTION 4**

- (Exam Topic 1)

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

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

- A. dev
- B. azurerm\_resource\_group
- C. azurerm
- D. test

**Answer:** A

**NEW QUESTION 5**

- (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 6**

- (Exam Topic 1)

Terraform can only manage resource dependencies if you set them explicitly with the depends\_on argument.

- A. True
- B. False

**Answer:** A

**Explanation:**

"Use the depends\_on meta-argument to handle hidden resource or module dependencies that Terraform cannot automatically infer. You only need to explicitly specify a dependency when a resource or module relies on another resource's behavior but does not access any of that resource's data in its arguments."  
[https://www.terraform.io/language/meta-arguments/depends\\_on](https://www.terraform.io/language/meta-arguments/depends_on)

**NEW QUESTION 7**

- (Exam Topic 1)

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

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

**Answer:** B

**Explanation:**

The variable declaration can also include a default argument.

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

**NEW QUESTION 8**

- (Exam Topic 1)

When running the command terraform taint against a managed resource you want to force recreation upon, Terraform will immediately destroy and recreate the resource.

- A. True
- B. False

**Answer:** B

**Explanation:**

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

**NEW QUESTION 9**

- (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 10

- (Exam Topic 1)

Which task does terraform init not perform?

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

**Answer:** D

### Explanation:

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

## NEW QUESTION 10

- (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 13

- (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 18

- (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 21**

- (Exam Topic 1)

A terraform apply can not \_\_\_\_\_ infrastructure.

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

**Answer:** D

**Explanation:**

<https://www.educative.io/answers/what-is-the-command-to-destroy-infrastructure-in-terraform>

**NEW QUESTION 25**

- (Exam Topic 1)

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

How can you do this safely?

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

**Answer:** A

**Explanation:**

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

**NEW QUESTION 29**

- (Exam Topic 1)

Only the user that generated a plan may apply it.

- A. True
- B. False

**Answer:** B

**NEW QUESTION 32**

- (Exam Topic 1)

You have declared a variable called `var.list` which is a list of objects that all have an attribute `id`. Which options will produce a list of the IDs? (Choose two.)

- A. `{ for o in var.list : o => o.id }`
- B. `var.list[*].id`
- C. `[ var.list[*].id ]`
- D. `[ for o in var.list : o.id ]`

**Answer:** BD

**Explanation:**

<https://www.terraform.io/language/expressions/splat>

A splat expression provides a more concise way to express a common operation that could otherwise be performed with a for expression.

**NEW QUESTION 34**

- (Exam Topic 1)

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

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

**Answer:** B

**NEW QUESTION 39**

- (Exam Topic 1)

You just scaled your VM infrastructure and realized you set the count variable to the wrong value. You correct the value and save your change.

What do you do next to make your infrastructure match your configuration?

- A. Run an apply and confirm the planned changes

- B. Inspect your Terraform state because you want to change it
- C. Reinitialize because your configuration has changed
- D. Inspect all Terraform outputs to make sure they are correct

**Answer:** A

#### NEW QUESTION 42

- (Exam Topic 1)

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

- A. for\_each
- B. repeated
- C. nesting
- D. dynamic

**Answer:** D

#### Explanation:

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

#### NEW QUESTION 43

- (Exam Topic 1)

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

- A. True
- B. False

**Answer:** A

#### Explanation:

Output values are like function return values.

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

#### NEW QUESTION 44

- (Exam Topic 1)

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

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

- A. element(aws\_instance.web, 2)
- B. aws\_instance.web[1].name
- C. aws\_instance.web[1]
- D. aws\_instance.web[2].name
- E. aws\_instance.web.\*.name

**Answer:** B

#### Explanation:

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

#### NEW QUESTION 46

- (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 48

- (Exam Topic 1)

You have declared an input variable called environment in your parent module. What must you do to pass the value to a child module in the configuration?

- A. Add node\_count = var.node\_count
- B. Declare the variable in a terraform.tfvars file



- C. Declare a node\_count input variable for child module
- D. Nothing, child modules inherit variables of parent module

**Answer:** C

**Explanation:**

"That module may call other modules and connect them together by passing output values from one to input values of another."  
<https://www.terraform.io/language/modules/develop>

**NEW QUESTION 51**

- (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 53**

- (Exam Topic 1)

A Terraform local value can reference other Terraform local values.

- A. True
- B. False

**Answer:** A

**Explanation:**

"The expressions in local values are not limited to literal constants; they can also reference other values in the module in order to transform or combine them, including variables, resource attributes, or other local values:" <https://www.terraform.io/language/values/locals#declaring-a-local-value>

**NEW QUESTION 57**

- (Exam Topic 1)

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

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

**Answer:** D

**Explanation:**

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

**NEW QUESTION 62**

- (Exam Topic 1)

Setting the TF\_LOG environment variable to DEBUG causes debug messages to be logged into syslog.

- A. True
- B. False

**Answer:** B

**Explanation:**

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

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

**NEW QUESTION 64**

- (Exam Topic 1)

Terraform and Terraform providers must use the same major version number in a single configuration.

- A. True
- B. False

**Answer:** B

**Explanation:**

<https://www.terraform.io/language/expressions/version-constraints#terraform-core-and-provider-versions>

**NEW QUESTION 65**

- (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 70**

- (Exam Topic 1)

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

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

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

**Answer:** C

**Explanation:**

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

**NEW QUESTION 72**

- (Exam Topic 1)

You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season. Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.

How would you solve this using infrastructure as code?

- A. Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource
- B. Implement a checklist that engineers can follow when configuring compute instances
- C. Replace the compute instance type with a larger version to reduce the number of required deployments
- D. Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews

**Answer:** D

**NEW QUESTION 75**

- (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 77**

- (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 82**

- (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 87**

- (Exam Topic 1)

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

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

**Answer: A**

**NEW QUESTION 92**

- (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 97

- (Exam Topic 1)

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

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

**Answer: A**

**Explanation:**

<https://www.terraform.io/cli/commands/refresh#:~:text=The%20terraform%20refresh%20command%20reads%>

**NEW QUESTION 98**

**NEW QUESTION**  
- (Exam Topic 1)

How is the Terraform remote backend different than other state backends such as S3, Consul, etc.?

- A. It can execute Terraform runs on dedicated infrastructure on premises or in Terraform Cloud
- B. It doesn't show the output of a terraform apply locally
- C. It is only available to paying customers
- D. All of the above

**Answer: A**

**Explanation:**

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

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

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

## NEW QUESTION 100

**NEW QUESTION**  
- (Exam Topic 1)

You have multiple team members collaborating on infrastructure as code (IaC) using Terraform, and want to apply formatting standards for readability.

How can you format Terraform HCL (HashiCorp Configuration Language) code according to standard Terraform style convention?

- A. Run the terraform fmt command during the code linting phase of your CI/CD process
- B. Designate one person in each team to review and format everyone's code
- C. Manually apply two spaces indentation and align equal sign "=" characters in every Terraform file (\*.tf)
- D. Write a shell script to transform Terraform files using tools such as AWK, Python, and sed

**Answer: A**

**Explanation:**

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

**NEW QUESTION 102**

- (Exam Topic 2)

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

- A. terraform graph -h
- B. terraform init
- C. terraform graph
- D. terraform fmt

**Answer:** D

**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.

**NEW QUESTION 107**

- (Exam Topic 2)

When TF\_LOG\_PATH is set, TF\_LOG must be set in order for any logging to be enabled.

- A. False
- B. True

**Answer:** B

**Explanation:**

TF\_LOG\_PATH specifies where the log should persist its output to. Note that even when TF\_LOG\_PATH is set, TF\_LOG must be set in order for any logging to be enabled.

For example, to always write the log to the directory you're currently running terraform from: export TF\_LOG\_PATH=./terraform.log  
export TF\_LOG=TRACE

**NEW QUESTION 110**

- (Exam Topic 2)

Which of the below are paid features of Terraform Cloud?

- A. Full API Coverage
- B. Secure variable Storage
- C. Roles/ Team management
- D. Cost Estimation
- E. Private Module Registry
- F. Sentinel policies

**Answer:** CDF

**Explanation:**

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

**NEW QUESTION 113**

- (Exam Topic 2)

While using generic git repository as a module source, which of the below options allows terraform to select a specific version or tag instead of selecting the HEAD.

- A. Append ref argument asmodule "vpc" { source = "git::https://example.com/vpc.git?ref=v1.2.0"}
- B. Append version argument asmodule "vpc" { source = "git::https://example.com/vpc.git?version=v1.2.0"}
- C. Append ref argument asmodule "vpc" { source = "git::https://example.com/vpc.git#ref=v1.2.0"}
- D. By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository and you can not override this.

**Answer:** A

**Explanation:**

By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository. You can override this using the ref argument:

```
module "vpc" {  
  source = "git::https://example.com/vpc.git?ref=v1.2.0"  
}
```

The value of the ref argument can be any reference that would be accepted by the git checkout command, including branch and tag names.

<https://www.terraform.io/docs/modules/sources.html>

**NEW QUESTION 116**

- (Exam Topic 2)

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

- A. False
- B. True

**Answer:** B

**Explanation:**

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

**NEW QUESTION 117**

- (Exam Topic 2)

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

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

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

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

**Answer:** A

**NEW QUESTION 119**

- (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 121**

- (Exam Topic 2)

Matt wants to import a manually created EC2 instance into terraform so that he can manage the EC2 instance through terraform going forward. He has written the configuration file of the EC2 instance before importing it to Terraform. Following is the code:

```
resource "aws_instance" "matt_ec2" { ami = "ami-bg2640de" instance_type = "t2.micro" vpc_security_group_ids = ["sg-6ae7d613", "sg-53370035"] key_name = "mysecret" subnet_id = "subnet-9e3cfbc5" }
```

The instance id of that EC2 instance is i-0260835eb7e9bd40 How he can import data of EC2 to state file?

- A. terraform import aws\_instance.id = i-0260835eb7e9bd40
- B. terraform import i-0260835eb7e9bd40
- C. terraform import aws\_instance.i-0260835eb7e9bd40
- D. terraform import aws\_instance.matt\_ec2 i-0260835eb7e9bd40

**Answer:** D

**Explanation:**

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

**NEW QUESTION 126**

- (Exam Topic 2)

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

- A. TF\_TRACE
- B. TF\_DEBUG
- C. TF\_LOG
- D. TF\_INFO

**Answer:** C

**Explanation:**

Terraform has detailed logs that can be enabled by setting the TF\_LOG environment variable to any value. This will cause detailed logs to appear on stderr.

You can set TF\_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF\_LOG is set to something other than a log level name. <https://www.terraform.io/docs/internals/debugging.html>

**NEW QUESTION 128**

- (Exam Topic 2)

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

choose?

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

**Answer: C**

**Explanation:**

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

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

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

**NEW QUESTION 130**

- (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 135**

- (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 138**

- (Exam Topic 2)

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

- A. False
- B. True

**Answer: B**

**Explanation:**

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

**NEW QUESTION 143**

- (Exam Topic 2)

lookup retrieves the value of a single element from which of the below data type?

- A. map
- B. set

- C. string
- D. list

**Answer:** A

**Explanation:**

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

**NEW QUESTION 145**

- (Exam Topic 2)

What is the default backend for Terraform?

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

**Answer:** C

**Explanation:**

By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to. <https://www.terraform.io/docs/backends/index.html>

**NEW QUESTION 147**

- (Exam Topic 2)

Which Terraform command will force a marked resource to be destroyed and recreated on the next apply?

- A. terraform fmt
- B. terraform destroy
- C. terraform taint
- D. terraform refresh

**Answer:** C

**Explanation:**

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

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

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

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

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

**NEW QUESTION 150**

- (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 152**

- (Exam Topic 2)

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



- A. True
- B. False

**Answer:** B

**Explanation:**

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

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

**NEW QUESTION 154**

- (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 157**

- (Exam Topic 2)

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

- A. True
- B. False

**Answer:** A

**Explanation:**

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

**NEW QUESTION 160**

- (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 164**

- (Exam Topic 2)

How does Terraform handle working with so many providers?

- A. Terraform ships with all of the plugins embedded in the Terraform binary.
- B. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in the configuration's working directory.
- C. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in a shared, system-wide plugins directory.
- D. Terraform allows you to select the providers you want to support during the Terraform installation process.

**Answer:** B

**Explanation:**

Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins, even the core types such as AWS and Heroku. Users of Terraform are able to write new plugins in order to support new functionality in Terraform.



**NEW QUESTION 167**

- (Exam Topic 2)

Which of the below configuration file formats are supported by Terraform? (Select TWO)

- A. Node
- B. JSON
- C. Go
- D. YAML
- E. HCL

**Answer:** BE

**Explanation:**

Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations. <https://www.terraform.io/docs/configuration/>

**NEW QUESTION 169**

- (Exam Topic 2)

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

- A. terraform providers
- B. terraform state show
- C. terraform provider
- D. terraform plan

**Answer:** A

**Explanation:**

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

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

**NEW QUESTION 170**

- (Exam Topic 2)

What is the command you can use to set an environment variable named "var1" of type String?

- A. export TF\_VAR\_VAR1
- B. set TF\_VAR\_var1
- C. variable "var1" { type = "string" }
- D. export TF\_VAR\_var1

**Answer:** D

**Explanation:**

The environment variable must be in the format TF\_VAR\_name, so for the QUESTION NO: TF\_VAR\_var1 is the correct choice.

[https://www.terraform.io/docs/commands/environment-variables.html#tf\\_var\\_name](https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name)

**NEW QUESTION 174**

- (Exam Topic 2)

What does terraform plan do ?

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

**Answer:** A

**NEW QUESTION 178**

- (Exam Topic 3)

Which of the below options is the equivalent Terraform 0.12 version of the snippet which is written in Terraform 0.11?

"\${var.instance\_id}"

- A. variable.instance\_id
- B. var.instance\_ids
- C. var.instance\_id
- D. None of the above

**Answer:** C

**NEW QUESTION 182**

- (Exam Topic 3)

Your manager has instructed you to start using terraform for the entire infra provisioning of the application stack. There are 4 environments – DEV , QA , UAT , and PROD. The application team has asked for complete segregation between these environments including the backend , state , and also configurations ,since there will be unique resources in different environments . What is the possible way to structure the terraform code to facilitate that.

- A. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to a different backend.
- B. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file ,

and map to the same backend.

C. Implement terraform workspaces , and map each environment with one workspace.

D. Enable remote backend storage . Configure 4 different backend storages , one for each environment.

**Answer:** A

**Explanation:**

In particular, organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g. staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are not a suitable isolation mechanism for this scenario.

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

**NEW QUESTION 187**

- (Exam Topic 3)

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

A. 5

B. 2

C. 3

D. 1

**Answer:** D

**Explanation:**

A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.

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

**NEW QUESTION 190**

- (Exam Topic 3)

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 191**

- (Exam Topic 3)

Which of the following state management command allow you to retrieve a list of resources that are part of the state file?

A. terraform state list

B. terraform state view

C. terraform view

D. terraform list

**Answer:** A

**Explanation:**

The terraform state list command is used to list resources within a Terraform state. Usage: terraform state list [options] [address...]

The command will list all resources in the state file matching the given addresses (if any). If no addresses are given, all resources are listed.

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

**NEW QUESTION 195**

- (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 200**

- (Exam Topic 3)

Which of the below options is a valid interpolation syntax for retrieving a data source?

A. \${google\_storage\_bucket.backend}

B. \${azurerm\_resource\_group.test.data}

C. \${aws\_instance.web.id.data}

D. \${data.google\_dns\_keys.foo\_dns\_keys.key\_signing\_keys[0].ds\_record}

**Answer:** D

**Explanation:**

Data source attributes are interpolated with the general syntax data.TYPE.NAME.ATTRIBUTE. The interpolation for a resource is the same but without the data. prefix (TYPE.NAME.ATTRIBUTE).

<https://www.terraform.io/docs/configuration-0-11/interpolation.html#attributes-of-a-data-source>

**NEW QUESTION 205**

- (Exam Topic 3)

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

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

**Answer:** A

**NEW QUESTION 209**

- (Exam Topic 3)

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

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

**Answer:** A

**Explanation:**

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

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

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

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

This example will taint a single resource:

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

The resource aws\_security\_group.allow\_all in the module root has been marked as tainted. <https://www.terraform.io/docs/commands/taint.html>

**NEW QUESTION 210**

- (Exam Topic 3)

Which of the following value will be accepted for var1? variable "var1" {  
type = string  
}

- A. None of the above
- B. Both A and B
- C. "5"
- D. 5

**Answer:** B

**Explanation:**

Terraform automatically converts number and bool values to strings when needed.

**NEW QUESTION 212**

- (Exam Topic 3)

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

- A. Repositories
- B. Workspaces
- C. Environment Variables
- D. Backends

**Answer:** B

**Explanation:**

workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. They are convenient in a number of situations, but cannot solve all problems.

A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. For example, a developer working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.

Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure. When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that

changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.

<https://www.terraform.io/docs/state/workspaces.html> <https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces>

**NEW QUESTION 215**

- (Exam Topic 3)

You have multiple developers working on a terraform project (using terraform OSS), and have saved the terraform state in a remote S3 bucket . However ,team is intermittently experiencing inconsistencies in the provisioned infrastructure / failure in the code . You have traced this problem to simultaneous/concurrent runs of terraform apply command for 2/more developers . What can you do to fix this problem?

- A. Use terraform workspaces feature, this will fix this problem by default , as every developer will have their own state file , and terraform will merge them on server side on its own.
- B. Structure your team in such a way that only one individual will run terraform apply , everyone will just make changes and share with hi
- C. Then there will be no chance of any inconsistencies.
- D. Stop using remote state , and store the developer tfstate in their own machine . Once a day , all developers should sit together and merge the state files manually , to avoid any inconsistencies.
- E. Enable terraform state locking for the S3 backend using DynamoDB tabl
- F. This prevents others from acquiring the lock and potentially corrupting your state.

**Answer:** D

**Explanation:**

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

**NEW QUESTION 220**

- (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 222**

- (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 224**

- (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 229**

- (Exam Topic 3)

You cannot publish your own modules on the Terraform Registry.

- A. False
- B. True

**Answer:** A

**Explanation:**

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

**NEW QUESTION 232**

- (Exam Topic 3)

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

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

**Answer:** B

**Explanation:**

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

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

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

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

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

**NEW QUESTION 237**

- (Exam Topic 3)

Which of the below commands will rename a EC2 instance without destroying and recreating it?

- A. terraform state mv
- B. terraform mv
- C. terraform plan
- D. terraform plan mv

**Answer:** A

**NEW QUESTION 242**

- (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 246**

- (Exam Topic 3)

Taint the resource "aws\_instance" "baz" resource that lives in module bar which lives in module foo.

- A. terraform taint module.foo.module.bar.baz
- B. terraform taint module.foo.bar.aws\_instance.baz
- C. terraform taint module.foo.module.bar.aws\_instance.baz
- D. terraform taint foo.bar.aws\_instance.baz

**Answer:** C

**Explanation:**

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

**NEW QUESTION 250**

- (Exam Topic 3)

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

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



F. Oracle Linux

**Answer:** ABCDEF

**Explanation:**

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

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

Debian 7.7+

Ubuntu 14.04.5 / 16.04 / 18.04

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

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

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

**NEW QUESTION 251**

- (Exam Topic 3)

Command terraform refresh will update state file?

A. False

B. True

**Answer:** B

**Explanation:**

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

This does not modify infrastructure, but does modify the state file. If the state is changed, this may cause changes to occur during the next plan or apply.

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

**NEW QUESTION 255**

- (Exam Topic 3)

You cannot publish your own modules on the Terraform Registry.

A. False

B. True

**Answer:** A

**Explanation:**

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

You have a Terraform configuration file where a variable itemNum is defined as follows: variable "itemNum" { default = 3}

**NEW QUESTION 259**

- (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 262**

- (Exam Topic 3)

The terraform state command can be used to \_\_\_\_\_

A. Update current state

B. Refresh existing state file

C. Print the current state file in console

D. It is not a valid command

**Answer:** A

**Explanation:**

The terraform state command is used for advanced state management. Rather than modify the state directly, the terraform state commands can be used in many cases instead.

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

**NEW QUESTION 263**



- (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 265

- (Exam Topic 3)

Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?

- A. Use data source aws\_ami.
- B. Create a map of region to ami id.
- C. Create different configuration file for different region.
- D. None of the above

**Answer:** A

**Explanation:**

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

#### NEW QUESTION 269

- (Exam Topic 3)

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

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

**Answer:** A

**Explanation:**

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

#### NEW QUESTION 273

- (Exam Topic 3)

Why is it a good idea to declare the required version of a provider in a Terraform configuration file?

- \* 1. terraform
- \* 2. {
- \* 3. required\_providers
- \* 4. {
- \* 5. aws = "~> 1.0"
- \* 6. }
- \* 7. }

- A. To remove older versions of the provider.
- B. To ensure that the provider version matches the version of Terraform you are using.
- C. Providers are released on a separate schedule from Terraform itself; therefore a newer version could introduce breaking changes.
- D. To match the version number of your application being deployed via Terraform.

**Answer:** C

#### NEW QUESTION 274

- (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 276**

- (Exam Topic 3)

A user has created three workspaces using the command line - prod, dev, and test. The user wants to create a fourth workspace named stage. Which command will the user execute to accomplish this?

- A. terraform workspace new stage
- B. terraform workspace -new stage
- C. terraform workspace -create stage
- D. terraform workspace create stage

**Answer:** A

**Explanation:**

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

**NEW QUESTION 278**

- (Exam Topic 4)

Which of the following is an invalid variable name?

- A. count
- B. web
- C. var1
- D. instance\_name

**Answer:** A

**Explanation:**

<https://www.terraform.io/intro/examples/count.html>

**NEW QUESTION 280**

- (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 282**

- (Exam Topic 4)

Using the terraform state rm command against a resource will destroy it.

- A. True
- B. False

**Answer:** B

**NEW QUESTION 285**

- (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 287**

- (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 288**

- (Exam Topic 4)

What is the purpose of a Terraform workspace in either open source or enterprise?

- A. Workspaces allow you to manage collections of infrastructure in state files.
- B. A logical separation of business units
- C. A method of grouping multiple infrastructure security policies
- D. Provides limited access to a cloud environment

**Answer:** B

**NEW QUESTION 292**

- (Exam Topic 4)

As a developer, you want to ensure your plugins are up to date with the latest versions. Which Terraform command should you use?

- A. terraform providers -upgrade
- B. terraform apply -upgrade
- C. terraform refresh -upgrade
- D. terraform init -upgrade

**Answer:** D

**NEW QUESTION 293**

- (Exam Topic 4)

Which one is the right way to import a local module named 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 294**

- (Exam Topic 4)

You have configured an Auto Scaling group in AWS to automatically scale the number of instances behind a load balancer based on the instances CPU utilization. The instances are configured using a Launch Configuration. You have observed that the Auto Scaling group doesn't successfully scale when you apply changes that require replacing the Launch Configuration. Why is this happening?

- A. You need to configure an explicit dependency for the Auto Scaling group using the depends\_on meta-parameter.
- B. You need to configure an explicit dependency for the Launch Configuration using the depends\_on meta-parameter.
- C. You need to configure the Auto Scaling group's create\_before\_destroy meta-parameter.
- D. You need to configure the Launch Configuration's create\_before\_destroy meta-parameter.

**Answer:** D

**Explanation:**

[https://www.terraform.io/docs/providers/aws/r/launch\\_configuration.html#using-with-autoscaling-groups](https://www.terraform.io/docs/providers/aws/r/launch_configuration.html#using-with-autoscaling-groups)

**NEW QUESTION 298**

- (Exam Topic 4)

Where can Terraform not load a provider from?

- A. Plugins directory
- B. Provider plugin cache
- C. Official HashiCorp distribution on releases, hashicorp.com
- D. Source code

**Answer:** D

**NEW QUESTION 303**

- (Exam Topic 4)

John is writing a module and within the module, there are multiple places where he has to use the same conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration. What is a better approach to dealing with

this?

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

**Answer:** A

**Explanation:**

A local value assigns a name to an expression, allowing it to be used multiple times within a module without repeating it.  
<https://www.terraform.io/docs/configuration/locals.html>

**NEW QUESTION 306**

- (Exam Topic 4)

Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)

- A. Explicit use of a provider block in configuration, optionally including a version constraint.
- B. Use of any resource belonging to a particular provider in a resource or data block in configuration.
- C. Existence of any resource instance belonging to a particular provider in the current state.
- D. Existence of any provider plugins found locally in the working directory.

**Answer:** ABC

**Explanation:**

The existence of a provider plugin found locally in the working directory does not itself create a provider dependency. The plugin can exist without any reference to it in the terraform configuration. <https://www.terraform.io/docs/commands/providers.html>

**NEW QUESTION 307**

- (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 311**

- (Exam Topic 4)

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

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

**Answer:** D

**NEW QUESTION 314**

- (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 316**

- (Exam Topic 4)

Which of the following statements best describes the Terraform list(...) type?

- A. a collection of values where each is identified by a string label.
- B. a sequence of values identified by consecutive whole numbers starting with zero.
- C. a collection of unique values that do not have any secondary identifiers or ordering.
- D. a collection of named attributes that each have their own type.

**Answer:** B

**Explanation:**

A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.  
<https://www.terraform.io/docs/configuration/types.html#structural-types>

**NEW QUESTION 320**

- (Exam Topic 4)

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

- A. False
- B. True

**Answer:** B

**NEW QUESTION 323**

- (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 324**

- (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 325**

- (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 330**

- (Exam Topic 4)

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

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

**Answer:** AC

**NEW QUESTION 335**

- (Exam Topic 4)

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

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

A)

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

B)

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

C)

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

A. Option A

B. Option B

C. Option C

**Answer: B**

**Explanation:**

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

**NEW QUESTION 337**

- (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 338**

- (Exam Topic 4)

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

- A. True
- B. False

**Answer:** B

**NEW QUESTION 342**

- (Exam Topic 4)

What is the result of the following terraform function call?

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

**Answer:** B

**Explanation:**

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

**NEW QUESTION 343**

- (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 346**

- (Exam Topic 4)

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

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

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

- A. terraform push
- B. terraform init
- C. terraform refresh
- D. terraform state

**Answer:** B

**NEW QUESTION 348**

- (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 350**

- (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 351**

- (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 355**

- (Exam Topic 4)

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

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

**Answer:** B

**Explanation:**

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

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

The Registry is available in two variants:

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

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

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

**NEW QUESTION 359**

- (Exam Topic 4)

colleague is new to Terraform and wants to add a new workspace named new-hire. What command he should execute from the following?

- A. terraform workspace-new-new-hire
- B. terraform workspace new new hire
- C. terraform workspace init new-hire
- D. terraform workspace new-hire

**Answer:** B

**NEW QUESTION 361**

- (Exam Topic 4)

How would you reference the Volume IDs associated with the ebs\_block\_device blocks in this configuration?

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

  ebs_block_device {
    device_name = "sda2"
    volume_size = 16
  }

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

- A. aws\_instance.example.ebs\_block\_device.[\*].volume\_id
- B. aws\_instance.example.ebs\_block\_device.volume\_id
- C. aws\_instance.example.ebs\_block\_device[sda2,sda3].volume\_id
- D. aws\_instance.example.ebs\_block\_device.\*.volume\_id

**Answer:** A

**Explanation:**

[https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/device\\_naming.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/device_naming.html)

**NEW QUESTION 362**

- (Exam Topic 4)

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

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

**Answer:** A

**Explanation:**

The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.

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

**NEW QUESTION 365**

- (Exam Topic 4)

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

- A. True
- B. False

**Answer:** B

**NEW QUESTION 370**

- (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 375**

- (Exam Topic 4)

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

- A. name
- B. alias
- C. depends\_on
- D. id

**Answer:** B

**NEW QUESTION 378**

- (Exam Topic 4)

Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

- A. Unlock
- B. force-unlock
- C. Removing the lock on a state file is not possible
- D. state-unlock

**Answer:** B

**Explanation:**

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

**NEW QUESTION 382**

- (Exam Topic 4)

How does Terraform determine dependencies between resources?

- A. Terraform automatically builds a resource graph based on resources, provisioners, special meta-parameters, and the state file, if present.
- B. Terraform requires all dependencies between resources to be specified using the depends\_on parameter
- C. Terraform requires resources in a configuration to be listed in the order they will be created to determine dependencies
- D. Terraform requires resource dependencies to be defined as modules and sourced in order

**Answer:** A

**Explanation:**

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

**NEW QUESTION 387**

- (Exam Topic 4)

You have modified your local Terraform configuration and ran terraform plan to review the changes. Simultaneously, your teammate manually modified the infrastructure component you are working on. Since you already ran terraform plan locally, the execution plan for terraform apply will be the same.

- A. True
- B. False

**Answer:** B

**NEW QUESTION 389**

- (Exam Topic 4)

Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.

What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

- A. Submit a ticket to AWS and ask them to export the state of all existing resources and use terraform import to import them into the state file.
- B. Delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward.
- C. Resources that are manually deployed in the AWS console cannot be imported by Terraform.
- D. Using terraform import, import the existing infrastructure into your Terraform state.

**Answer:** D

**Explanation:**

Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.

This is a great way to slowly transition infrastructure to Terraform.

The terraform import command is used to import existing infrastructure.

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

Example:

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

```
}
```

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

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

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

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

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

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside

Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws\_instance.import\_example in the Terraform state.

### NEW QUESTION 393

- (Exam Topic 4)

All Terraform Cloud tiers support team management and governance.

A. True

B. False

**Answer: B**

#### Explanation:

<https://www.terraform.io/cloud-docs/overview>

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. Each higher paid upgrade plan is a strict superset of any lower plans — for example, the Team & Governance plan includes all of the features of the Team plan.

### NEW QUESTION 398

- (Exam Topic 4)

You have written a terraform IaC script which was working till yesterday , but is giving some vague error from today , which you are unable to understand . You want more detailed logs that could potentially help you troubleshoot the issue , and understand the root cause. What can you do to enable this setting? Please note , you are using terraform OSS.

A. Terraform OSS can push all its logs to a syslog endpoint

B. As such, you have to set up the syslog sink, and enable TF\_LOG\_PATH env variable to the syslog endpoint and all logs will automatically start streaming.

C. Detailed logs are not available in terraform OSS, except the crash messag

D. You need to upgrade to terraform enterprise for this point.

E. Enable the TF\_LOG\_PATH to the log sink file location, and logging output will automatically be stored there.

F. Enable TF\_LOG to the log level DEBUG, and then set TF\_LOG\_PATH to the log sink file location. Terraform debug logs will be dumped to the sink path, even in terraform OSS.

**Answer: D**

#### Explanation:

Terraform has detailed logs which can be enabled by setting the TF\_LOG environment variable to any value. This will cause detailed logs to appear on stderr.

You can set TF\_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF\_LOG is set to something other than a log level name.

To persist logged output you can set TF\_LOG\_PATH in order to force the log to always be appended to a specific file when logging is enabled. Note that even when TF\_LOG\_PATH is set, TF\_LOG must be set in order for any logging to be enabled.

### NEW QUESTION 402

- (Exam Topic 4)

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

A. True

B. False

**Answer: B**

#### Explanation:

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

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

### NEW QUESTION 403

- (Exam Topic 4)

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

A. True

B. False

**Answer: B**

### NEW QUESTION 407

- (Exam Topic 4)

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

A. Before the plan phase

B. During the plan phase

C. Before the a apply phase



D. After the apply phase

**Answer:** C

**Explanation:**

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

**NEW QUESTION 409**

- (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 412**

- (Exam Topic 4)

Why does this backend configuration not follow best practices?

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

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

**Answer:** A

**NEW QUESTION 413**

- (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 417**

- (Exam Topic 4)

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

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

**Answer:** ABDEF

**Explanation:**

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

**NEW QUESTION 421**

- (Exam Topic 4)

What kind of configuration block will create an infrastructure object with settings specified in the block?

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

**Answer:** C

**NEW QUESTION 425**

- (Exam Topic 4)

When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.

For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.

- A. Export the value using terraform export and input the value using terraform input.
- B. Configure the pertinent provider's configuration with a list of possible IP addresses to use.
- C. Configure an output value in the application module in order to use that value for the DNS module.
- D. Preconfigure the IP address as a parameter in the DNS module.

**Answer:** C

**Explanation:**

Output values are like the return values of a Terraform module, and have several uses:

- \* A child module can use outputs to expose a subset of its resource attributes to a parent module.
- \* A root module can use outputs to print certain values in the CLI output after running terraform apply.
- \* When using remote state, root module outputs can be accessed by other configurations via a terraform\_remote\_state data source.

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

**NEW QUESTION 427**

- (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 431**

- (Exam Topic 4)

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

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

**Answer:** C

**Explanation:**

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

**NEW QUESTION 434**

- (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 435

- (Exam Topic 4)

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

- A. terraform inspect
- B. terraform read
- C. terraform show
- D. terraform state

**Answer: C**

#### NEW QUESTION 438

- (Exam Topic 4)

You want to share Terraform state with your team, store it securely and provide state locking. How would you do this? Choose three correct answers.

- A. Using the consul Terraform backend.
- B. Using the remote Terraform backend with Terraform Cloud / Terraform Enterprise.
- C. Using the local backend.
- D. Using the s3 terraform backen
- E. The dynamodb\_field option e not needed.
- F. Using an s3 terraform backend with an appropriate IAM policy and dynamodb\_field option configured.

**Answer: ABE**

#### NEW QUESTION 443

- (Exam Topic 4)

What is a downside to using the Vault provider to read secrets from Vault?

- A. Secrets are persisted to the state file and plans.
- B. Terraform and Vault must be running on the same version.
- C. Terraform and Vault must be running on the same physical host.
- D. Terraform requires a unique auth method to work with Vault.

**Answer: A**

#### Explanation:

The Vault provider allows Terraform to read from, write to, and configure Hashicorp Vault.

Interacting with Vault from Terraform causes any secrets that you read and write to be persisted in both Terraform's state file and in any generated plan files. For any Terraform module that reads or writes Vault secrets, these files should be treated as sensitive and protected accordingly.

#### NEW QUESTION 445

- (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 446**

- (Exam Topic 4)

Which provider authentication method prevents credentials from being stored in the state file?

- A. Using environment variables
- B. Specifying the login credentials in the provider block
- C. Setting credentials as Terraform variables
- D. None of the above

**Answer:** A

**NEW QUESTION 447**

- (Exam Topic 4)

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

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

**Answer:** D

**Explanation:**

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

**NEW QUESTION 451**

- (Exam Topic 4)

terraform validate reports HCL syntax errors.

- A. True
- B. False

**Answer:** A

**NEW QUESTION 455**

- (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 459**

- (Exam Topic 4)

A variable az has the following default value. What will be the datatype of the variable? az=["us-west-1a","us-east-1a"]

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

**Answer:** B

**NEW QUESTION 464**

- (Exam Topic 4)

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

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

**Answer:** A

**NEW QUESTION 465**

- (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 466

- (Exam Topic 4)

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

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

**Answer:** BCDE

**Explanation:**

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

#### NEW QUESTION 471

- (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 475

- (Exam Topic 4)

Terraform Enterprise (also referred to as pTFE) requires what type of backend database for a clustered deployment?

- A. PostgreSQL
- B. Cassandra
- C. MySQL
- D. MSSQL

**Answer:** A

**Explanation:**

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

#### NEW QUESTION 477

- (Exam Topic 4)

Which of the following is not valid source path for specifying a module?

- A. source = "./module?version=v1.0.0"
- B. source = "github.com/hashicorp/example?ref=v1.0.0"
- C. source = "./module"
- D. source = "hashicorp/consul/aws"

**Answer:** A

#### NEW QUESTION 481

- (Exam Topic 4)

Which Terraform collection type should you use to store key/value pairs?

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

**Answer:** D

**Explanation:**

Maps/objects are represented by a pair of curly braces containing a series of <KEY> = <VALUE> pairs Source:

<https://www.terraform.io/language/expressions/types>

**NEW QUESTION 486**

- (Exam Topic 4)

You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration? Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

**NEW QUESTION 491**

- (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 493**

- (Exam Topic 4)

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

- A. False
- B. True

**Answer:** B

**NEW QUESTION 496**

- (Exam Topic 4)

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

- A. a private install
- B. disconnected
- C. air-gapped
- D. non-traditional

**Answer:** D

**Explanation:**

A Terraform Enterprise install that is provisioned on a network that does not have Internet access is generally known as an air-gapped install. These types of installs require you to pull updates, providers, etc. from external sources vs. being able to download them directly.

**NEW QUESTION 501**

- (Exam Topic 4)

terraform apply is failing with the following error. What next step should you take to determine the root cause of the problem?  
Error loading state: AccessDenied: Access Denied status code: 403, request id: 288766CE5CCA24A0, host id: FOOBAR

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

**Answer:** A

**Explanation:**

Terraform has detailed logs which can be enabled by setting the TF\_LOG environment variable to any value. This will cause detailed logs to appear on stderr. You

can set TF\_LOG to one of the log levels (in order of decreasing verbosity) TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.

**NEW QUESTION 505**

- (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 510**

- (Exam Topic 4)

What Terraform feature is shown in the example below?

- A. conditional expression
- B. local values
- C. dynamic block
- D. data source

**Answer:** C

**NEW QUESTION 515**

- (Exam Topic 4)

Which of the below backends support state locking?

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

**Answer:** ABC

**NEW QUESTION 519**

- (Exam Topic 4)

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

- A. 1
- B. 2
- C. 4

**Answer:** C

**Explanation:**

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

Indent two spaces for each nesting level.

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

```
ami = "abc123" instance_type = "t2.micro"
```

When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them.

Use one blank line to separate the arguments from the blocks.

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

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

```
resource "aws_instance" "example" { count = 2 # meta-argument first
```

```
ami = "abc123" instance_type = "t2.micro" network_interface {
```

```
# ...
```

```
}
```

```
lifecycle { # meta-argument block last create_before_destroy = true
```

```
}
```

```
}
```

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

Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example: root\_block\_device, ebs\_block\_device and ephemeral\_block\_device on aws\_instance form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

**NEW QUESTION 520**

- (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 525

- (Exam Topic 4)

What does terraform refresh modify?

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

**Answer:** B

#### Explanation:

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

#### NEW QUESTION 528

- (Exam Topic 4)

Most Terraform providers interact with \_\_\_\_\_.

- A. API
- B. VCS Systems
- C. Shell scripts
- D. None of the above

**Answer:** A

#### Explanation:

Terraform relies on plugins called "providers" to interact with cloud providers, SaaS providers, and other APIs, as per: <https://www.terraform.io/language/providers>

#### NEW QUESTION 533

- (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 536

- (Exam Topic 4)

What does terraform destroy do?

- A. Destroy all infrastructure in the Terraform state file
- B. Destroy all Terraform code files in the current directory while leaving the state file intact
- C. Destroy all infrastructure in the configured Terraform provider
- D. Destroy the Terraform state file while leaving infrastructure intact

**Answer:** A

#### Explanation:

The terraform destroy command terminates resources managed by your Terraform project. This command is the inverse of terraform apply in that it terminates all the resources specified in your Terraform state. It does not destroy resources running elsewhere that are not managed by the current Terraform project.

<https://learn.hashicorp.com/tutorials/terraform/aws-destroy>

#### NEW QUESTION 540

- (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 545

- (Exam Topic 4)

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

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

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

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

A. Terraform will not make any changes to the Load Balancer and will update the state file to reflect any changes made.

B. Terraform will change the port back to 80 in your code

C. Terraform will change the load balancer port to 80, and then change it back to 443

D. Terraform will fail with an error because the state file is no longer accurate

**Answer: A**

#### **NEW QUESTION 549**

.....

## Thank You for Trying Our Product

\* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

\* One year free update

You can enjoy free update one year. 24x7 online support.

\* Trusted by Millions

We currently serve more than 30,000,000 customers.

\* Shop Securely

All transactions are protected by VeriSign!

**100% Pass Your TA-002-P Exam with Our Prep Materials Via below:**

<https://www.certleader.com/TA-002-P-dumps.html>