



# HashiCorp

## Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

#### NEW QUESTION 1

- (Exam Topic 1)

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

- A. terraform workspace
- B. terraform workspace show
- C. terraform workspace list
- D. terraform show workspace

**Answer: C**

#### Explanation:

terraform workspace list

The command will list all existing workspaces.

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

#### NEW QUESTION 2

- (Exam Topic 1)

You 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 3

- (Exam Topic 1)

When should you use the force-unlock command?

- A. You see a status message that you cannot acquire the lock
- B. You have a high priority change
- C. Automatic unlocking failed
- D. Your apply failed due to a state lock

**Answer: C**

#### Explanation:

Be very careful with this command. 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. Source: <https://www.terraform.io/language/state/locking>

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

#### 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)

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

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

**Answer: D**

#### Explanation:

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

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

#### NEW QUESTION 6

- (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 7

- (Exam Topic 1)

Terraform provisioners that require authentication can use the \_\_\_\_\_ block.

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

**Answer:** A

#### Explanation:

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

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

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

#### NEW QUESTION 8

- (Exam Topic 1)

Where does the Terraform local backend store its state?

- A. In the `/tmp` directory
- B. In the `terraform.tfvars` file
- C. In the `terraform.tfstate` file
- D. In the user's `.terraformrc` file

**Answer:** C

#### Explanation:

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

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

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

#### NEW QUESTION 9

- (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 10

- (Exam Topic 1)

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

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

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

**Answer:** D

**Explanation:**

version = ">= 1.2.0, < 2.0.0"

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

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

The following operators are valid:

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

!=: Excludes an exact version number.

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

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

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

**NEW QUESTION 10**

- (Exam Topic 1)

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

A. True

B. False

**Answer: B**

**Explanation:**

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

**NEW QUESTION 13**

- (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 18**

- (Exam Topic 1)

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

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

A. compute\_instance

B. main

C. google

D. test

**Answer: B**

**NEW QUESTION 20**

- (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 23**

- (Exam Topic 1)

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

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

A. Run terraform output ip\_address to view the result

B. In a new folder, use the terraform\_remote\_state data source to load in the state file, then write an output for each resource that you find the state file

- C. Run terraform state list to find the name of the resource, then terraform state show to find the attributes including public IP address
- D. Run terraform destroy then terraform apply and look for the IP address in stdout

**Answer:** C

**Explanation:**

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

**NEW QUESTION 27**

- (Exam Topic 1)

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

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

**Answer:** A

**Explanation:**

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

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

**NEW QUESTION 30**

- (Exam Topic 1)

What command does Terraform require the first time you run it within a configuration directory?

- A. terraform import
- B. terraform init
- C. terraform plan
- D. terraform workspace

**Answer:** B

**Explanation:**

terraform init command is used to initialize a working directory containing Terraform configuration files. Reference:

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

**NEW QUESTION 34**

- (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 39**

- (Exam Topic 2)

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

- A. Run terraform refresh to update the terraform state , then write the terraform code , and finally run terraform apply.
- B. Run terraform destroy first since you need to start from fresh every time , before running terraform apply.
- C. Write terraform code , and run terraform push , to update the terraform state to the remote repo , which in turn will take care of the next steps.
- D. Write the terraform code on the developer machine , run terraform plan to check the changes , and run terraform apply to provision the infra.

**Answer:** D

**Explanation:**

You do not need to run terraform refresh as terraform plan implicitly will run terraform refresh. <https://www.terraform.io/guides/core-workflow.html>

**NEW QUESTION 41**

- (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 45**

- (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 47**

- (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 48**

- (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 49**

- (Exam Topic 2)

Terraform init can indeed be run only a few times, because, every time terraform init will initialize the project

, and download all plugins from the internet repository , regardless of whether they were present or not , and this increases the waiting time

A. True

B. False

**Answer:** B

**Explanation:**

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

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

**NEW QUESTION 53**

- (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 56**

- (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 58**

- (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 60**

- (Exam Topic 2)

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

A. First Time Configuration

B. Default Configuration

C. Changing Configuration

D. Partial Configuration

E. Incomplete Configuration

**Answer:** D

**Explanation:**

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

With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments:

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

\* File: A configuration file may be specified via the init command line. To specify a file, use the

-backend-config=PATH option when running terraform init. If the file contains secrets it may be kept in a secure data store, such as Vault, in which case it must be downloaded to the local disk before running Terraform.

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

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

**NEW QUESTION 63**

- (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 67**

- (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 70**

- (Exam Topic 3)

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

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

**Answer:** D

**NEW QUESTION 73**

- (Exam Topic 3)

Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.

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

**Answer:** B

**Explanation:**

Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster.  
By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar.  
Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise. Terraform Cloud's private module registry helps you share Terraform modules across your organization.  
<https://www.terraform.io/docs/cloud/registry/index.html> <https://www.terraform.io/docs/cloud/registry/publish.html>

**NEW QUESTION 76**

- (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 80

- (Exam Topic 3)

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

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

**Answer:** A

#### Explanation:

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

#### NEW QUESTION 83

- (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 87

- (Exam Topic 3)

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.

- A. False
- B. True

**Answer:** B

#### Explanation:

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.

<https://www.terraform.io/docs/state/sensitive-data.html#recommendations>

#### NEW QUESTION 92

- (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 96

- (Exam Topic 3)

Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance. Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

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

**Answer:** B

**Explanation:**

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

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 97**

- (Exam Topic 3)

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

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

**Answer:** C

**Explanation:**

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

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

**NEW QUESTION 98**

- (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 102**

- (Exam Topic 3)

Multiple providers can be declared within a single Terraform configuration file.

- A. True
- B. False

**Answer:** A

**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>

**NEW QUESTION 105**

- (Exam Topic 3)

When using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects. Which of the below option is a recommended solution for this?

- A. Remote State
- B. Module
- C. Use the cached state and treat this as the record of truth.

D. Workspace

**Answer:** A

**Explanation:**

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

#### NEW QUESTION 107

- (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 109

- (Exam Topic 3)

Refer to the following terraform variable definition

```
variable "track_tag" { type = list default = ["data_ec2","integration_ec2","digital_ec2"]} track_tag = { Name = element(var.track_tag,count.index)}
```

If `count.index` is set to 2, which of the following values will be assigned to the `name` attribute of `track_tag` variable?

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

**Answer:** B

#### NEW QUESTION 114

- (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 115

- (Exam Topic 4)

terraform init retrieves the source code for all referenced modules

- A. True
- B. False

**Answer:** A

**Explanation:**

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

#### NEW QUESTION 119

- (Exam Topic 4)

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

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

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

- A. Yes
- B. No

**Answer:** A

**Explanation:**

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

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

**Limitations of Sensitive Outputs**

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

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

**NEW QUESTION 123**

- (Exam Topic 4)

Select the answer below that completes the following statement: Terraform Cloud can be managed from the CLI but requires \_\_\_\_\_?

- A. an API token
- B. a TOTP token
- C. a username and password
- D. authentication using MFA

**Answer:** A

**Explanation:**

API and CLI access are managed with API tokens, which can be generated in the Terraform Cloud UI. Each user can generate any number of personal API tokens, which allow access with their own identity and permissions. Organizations and teams can also generate tokens for automating tasks that aren't tied to an individual user.

**NEW QUESTION 127**

- (Exam Topic 4)

Terraform is currently being used by your organisation to create resources on AWS for the development of a web application. One of your coworkers wants to change the instance type to "t2.large" while keeping the default set values.

What adjustments does the teammate make in order to meet his goal?

- A. Issue Terraform plan instance.type".t2.large" and it deploys the instance
- B. Modify the tf.variables with the instance type and issue terraform apply
- C. Create a new file my.tfvars and add the type of the instance and issue terraform plan and apply
- D. Modify the terraform.tfvars with the instance type and issue terraform plan and then terraform apply to deploy the instances

**Answer:** D

**NEW QUESTION 129**

- (Exam Topic 4)

In the example below, the depends\_on argument creates what type of dependency?

- A. implicit dependency
- B. internal dependency
- C. explicit dependency
- D. non-dependency resource

**Answer:** C

**NEW QUESTION 134**

- (Exam Topic 4)

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

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

- A. resource.kubrnetes\_namespace>example.name
- B. kubernetes\_namespace.test.name
- C. kubernetes\_namespace.example.name
- D. data kubernetes\_namespace.name
- E. None of the above

**Answer:** C

**Explanation:**

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

**NEW QUESTION 139**

- (Exam Topic 4)

True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.

- A. False
- B. True

**Answer:** A

**Explanation:**

Currently, Terraform has no mechanism to redact or protect secrets that are returned via data sources, so secrets read via this provider will be persisted into the Terraform state, into any plan files, and in some cases in the console output produced while planning and applying. These artifacts must, therefore, all be protected accordingly.

**NEW QUESTION 143**

- (Exam Topic 4)

Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 ...) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region. What should you do? Your provider setting is as below  
# The default provider configuration provider "aws" { region = "us-east-1" }

- A. No way to enable this via a single script . Write 2 different scripts with different default providers in the 2 scripts , one for us-east , another for us-west.
- B. Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.
- C. Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).
- D. Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.

**Answer:** C

**Explanation:**

You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

```
# The default provider configuration provider "aws" {  
  region = "us-east-1"  
}  
# Additional provider configuration for west coast region provider "aws" {  
  alias = "west" region = "us-west-2"  
}
```

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

**NEW QUESTION 147**

- (Exam Topic 4)

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

- A. False
- B. True

**Answer:** A

**Explanation:**

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

**NEW QUESTION 148**

- (Exam Topic 4)

Terra form installs its providers during which phase?

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

**Answer:** B

**Explanation:**

Providers are installed in the init phase

**NEW QUESTION 153**

- (Exam Topic 4)

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

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

**Answer:** ACDF



**Explanation:**

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

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

**NEW QUESTION 157**

- (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 162**

- (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 163**

- (Exam Topic 4)

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

- A. False
- B. True

**Answer: B**

**NEW QUESTION 164**

- (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 167**

- (Exam Topic 4)

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

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

**Answer: CE**

**NEW QUESTION 168**

- (Exam Topic 4)

From the code below, identify the implicit dependency:

- A. The EIP with an id of ami-2757f631
- B. The AMI used for the EC2 instance
- C. The EC2 instance labeled web\_server
- D. The S3 bucket labeled company\_data



**Answer:** C

#### NEW QUESTION 170

- (Exam Topic 4)

Talal is a DevOps engineer and he has deployed the production infrastructure using Terraform. He is using a very large configuration file to maintain and update the actual infrastructure. As the infrastructure have grown to a very complex and large, he has started experiencing slowness when he run runs terraform plan. What are the options for him to resolve this slowness?

- A. Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.
- B. Run terraform refresh every time before running terraform plan.
- C. Break large configurations into several smaller configurations that can each be independently applied.
- D. Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.

**Answer:** CD

#### Explanation:

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

Although 'Use -refresh=false flag as well as the -target flag with terraform plan in order to work around this.' is a solution, but its not always recommended. Instead of using -target as a means to operate on isolated portions of very large configurations, prefer instead to break large configurations into several smaller configurations that can each be independently applied. Data sources can be used to access information about resources created in other configurations, allowing a complex system architecture to be broken down into more manageable parts that can be updated independently.

Option 'Run terraform refresh every time before running terraform plan.' and 'Use -refresh=true flag as well as the -target flag with terraform plan in order to work around this.' is not correct because in both the cases terraform will query every resources of the infrastructure.

#### NEW QUESTION 175

- (Exam Topic 4)

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

- A. True
- B. False

**Answer:** A

#### Explanation:

Initialization

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

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

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

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

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

#### NEW QUESTION 176

- (Exam Topic 4)

Why might a user opt to include the following snippet in their configuration file?

- A. Terraform 0.12 introduced substantial changes to the syntax used to write Terraform configuration
- B. The user wants to ensure that the application being deployed is a minimum version of 0.12
- C. this ensures that all Terraform providers are above a certain version to match the application being deployed
- D. versions before Terraform 0.12 were not approved by HashiCorp to be used in production

**Answer:** A

#### NEW QUESTION 177

- (Exam Topic 4)

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

- A. True
- B. False

**Answer:** B

#### Explanation:

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

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

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

#### NEW QUESTION 181

- (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 186

- (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 189

- (Exam Topic 4)

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

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

**Answer:** C

#### NEW QUESTION 190

- (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 193

- (Exam Topic 4)

A single terraform resource file that defines an aws\_instance resource can simple be renamed to azurerm\_virtual\_machine in order to switch cloud providers

- A. True
- B. False

**Answer:** B

#### Explanation:

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

#### NEW QUESTION 195

- (Exam Topic 4)

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

- A. False
- B. True

**Answer:** B

**Explanation:**

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

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

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

**NEW QUESTION 196**

- (Exam Topic 4)

Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)

- A. GitHub
- B. CVS Version Control
- C. Azure DevOps Server
- D. Bitbucket Cloud
- E. GitHub Enterprise

**Answer:** ACDE

**Explanation:**

Terraform Cloud supports the following VCS providers:

- <https://www.terraform.io/docs/cloud/vcs/github.html>
- <https://www.terraform.io/docs/cloud/vcs/github.html>
- <https://www.terraform.io/docs/cloud/vcs/github-enterprise.html>
- <https://www.terraform.io/docs/cloud/vcs/gitlab-com.html>
- <https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html>
- <https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html>
- <https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html>
- <https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html>
- <https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html> <https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers>

**NEW QUESTION 197**

- (Exam Topic 4)

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

- A. tfvar
- B. depends\_on
- C. instance aws
- D. varl

**Answer:** B

**NEW QUESTION 200**

- (Exam Topic 4)

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

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

**Answer:** D

**NEW QUESTION 203**

- (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 204**

- (Exam Topic 4)

You cannot install third party plugins using terraform init.

- A. True
- B. False

**Answer:** B

**Explanation:**

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

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

#### NEW QUESTION 208

- (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 209

- (Exam Topic 4)

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 terraform apply again immediately afterwards without changing any Terraform code?

- A. Terraform will terminate and recreate the VM
- B. Terraform will create another duplicate VM
- C. Terraform will apply the VM to the state file
- D. Nothing

**Answer:** D

#### NEW QUESTION 213

- (Exam Topic 4)

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

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

**Answer:** D

#### Explanation:

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

#### NEW QUESTION 215

- (Exam Topic 4)

In the below configuration, how would you reference the module output vpc\_id ?

```
module "vpc" {  
  source = "terraform-aws-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:

module.vpc.id

#### NEW QUESTION 217

- (Exam Topic 4)

You need to specify a dependency manually. What resource meta-parameter can you use to make sure Terraform respects the dependency? Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

- A. Mastered
- B. Not Mastered

**Answer:** A

#### Explanation:

depends\_on

#### NEW QUESTION 222

- (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 225

- (Exam Topic 4)

Which command lets you experiment with Terraform's built-in functions?

- A. terraform env
- B. terraform console
- C. terraform test
- D. terraform validate

**Answer:** B

**Explanation:**

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

#### NEW QUESTION 229

- (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 231

- (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 233

- (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 237

- (Exam Topic 4)

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

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

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

**Answer:** B

**Explanation:**

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

**NEW QUESTION 239**

- (Exam Topic 4)

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

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

**Answer:** D

**Explanation:**

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

**NEW QUESTION 242**

- (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 245**

- (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 248**

- (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 253**

- (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 255**

.....

## Thank You for Trying Our Product

### We offer two products:

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

2nd - Questons and Answers in PDF Format

### TA-002-P Practice Exam Features:

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

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