

HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate



NEW QUESTION 1

- (Exam Topic 1)

Which of the following is available only in Terraform Enterprise or Cloud workspaces and not in Terraform CLI?

- A. Secure variable storage
- B. Support for multiple cloud providers
- C. Dry runs with terraform plan
- D. Using the workspace as a data source

Answer: A

Explanation:

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

NEW QUESTION 2

- (Exam Topic 1)

A fellow developer on your team is asking for some help in refactoring their Terraform code. As part of their application's architecture, they are going to tear down an existing deployment managed by Terraform and deploy new. However, there is a server resource named `aws_instance.ubuntu[1]` they would like to keep to perform some additional analysis.

What command should be used to tell Terraform to no longer manage the resource?

- A. `terraform apply rm aws_instance.ubuntu[1]`
- B. `terraform state rm aws_instance.ubuntu[1]`
- C. `terraform plan rm aws_instance.ubuntu[1]`
- D. `terraform delete aws_instance.ubuntu[1]`

Answer: B

Explanation:

"You can use `terraform state rm` in the less common situation where you wish to remove a binding to an existing remote object without first destroying it, which will effectively make Terraform "forget" the object while it continues to exist in the remote system." <https://www.terraform.io/cli/commands/state/rm>

NEW QUESTION 3

- (Exam Topic 1)

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

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

Answer: C

Explanation:

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

NEW QUESTION 4

- (Exam Topic 1)

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 5

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 6

- (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 7

- (Exam Topic 1)

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

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

Answer: B

Explanation:

"The name of a variable can be any valid identifier except the following: source, version, providers, count, for_each, lifecycle, depends_on, locals."

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

NEW QUESTION 8

- (Exam Topic 1)

You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the `gcloud` command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.

What are the two things you must do to achieve this? (Choose two.)

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

Answer: BC

Explanation:

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

NEW QUESTION 9

- (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 10

- (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 10

- (Exam Topic 1)

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

- A. FTP server

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

Answer: A

Explanation:

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

NEW QUESTION 11

- (Exam Topic 1)

Which provisioner invokes a process on the resource created by Terraform?

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

Answer: A

Explanation:

"The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource."

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

"The remote-exec provisioner invokes a script on a remote resource after it is created." <https://www.terraform.io/language/resources/provisioners/remote-exec>

NEW QUESTION 13

- (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 14

- (Exam Topic 1)

Terraform validate reports syntax check errors from which of the following scenarios?

- A. Code contains tabs indentation instead of spaces
- B. There is missing value for a variable
- C. The state files does not match the current infrastructure
- D. None of the above

Answer: B

Explanation:

The terraform validate command is used to validate the syntax of the terraform files. Terraform performs a syntax check on all the terraform files in the directory, and will display an error if any of the files doesn't validate. This command does not check formatting (e.g. tabs vs spaces, newlines, comments etc.). The following can be reported: invalid HCL syntax (e.g. missing trailing quote or equal sign) invalid HCL references (e.g. variable name or attribute which doesn't exist) same provider declared multiple times same module declared multiple times same resource declared multiple times invalid module name interpolation used in places where it's unsupported (e.g. variable, depends_on, module.source, provider) missing value for a variable (none of -var foo=... flag, -var-file=foo.vars flag, TF_VAR_foo environment variable, terraform.tfvars, or default value in the configuration) <https://www.typeerror.org/docs/terraform/commands/validate>
<https://learning-ocean.com/tutorials/terraform/terraform-validate>

NEW QUESTION 19

- (Exam Topic 1)

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

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

Answer: D

Explanation:

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

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

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

NEW QUESTION 21

- (Exam Topic 1)

Module variable assignments are inherited from the parent module and do not need to be explicitly set.

- A. True
- B. False

Answer: B

NEW QUESTION 24

- (Exam Topic 1)

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

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

Answer: B

Explanation:

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

NEW QUESTION 26

- (Exam Topic 1)

You would like to reuse the same Terraform configuration for your development and production environments with a different state file for each. Which command would you use?

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

Answer: B

Explanation:

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

NEW QUESTION 27

- (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 31

- (Exam Topic 1)

What is the workflow for deploying new infrastructure with Terraform?

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

Answer: D

Explanation:

Reference: <https://www.google.com/search?q=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&aq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&aqs=chrome..69i57.556j0j7&sourceid=chrome&ie=UTF-8>

NEW QUESTION 32

- (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 33

- (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 34

- (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 39

- (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 40

- (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 43

- (Exam Topic 1)

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

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

Answer: A

NEW QUESTION 46

- (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 50

- (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 51

- (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 56

- (Exam Topic 1)

How is terraform import run?

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

Answer: D

Explanation:

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

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

NEW QUESTION 58

- (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 63

- (Exam Topic 1)

Which backend does the Terraform CLI use by default?

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

Answer: D

Explanation:

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

NEW QUESTION 65

- (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 68

- (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 72

- (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 73

- (Exam Topic 2)

Provisioners should only be used as a last resort.

- A. False
- B. True

Answer: B

Explanation:

Provisioners are a Last Resort

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

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

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

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

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

NEW QUESTION 77

- (Exam Topic 2)

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

- A. False
- B. True

Answer: A

Explanation:

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

NEW QUESTION 78

- (Exam Topic 2)

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

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

Answer: C

Explanation:

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

NEW QUESTION 79

- (Exam Topic 2)

Which one of the following will run `echo 0` and `echo 1` on a newly created host?

- A. `provisioner "local-exec" { command = "echo 0" command = "echo 1" }`
- B. `provisioner "remote-exec" { inline = [echo 0,echo 1]}`
- C. `provisioner "remote-exec" {command = "${echo 0}" command = "${echo 1}"}`
- D. `provisioner "remote-exec" { inline = ["echo 0","echo 1"]}`

Answer: D

Explanation:

remote-exec Provisioner Example usage

```
resource "aws_instance" "web" {
```

```
# ...
```

```
provisioner "remote-exec" { inline = [
```

```
"puppet apply",
```

```
"consul join ${aws_instance.web.private_ip}",
```

```
]
```

```
}
```

```
}
```

NEW QUESTION 81

- (Exam Topic 2)

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

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

Answer: A

Explanation:

IBM Cloud does not have a provider maintained by HashiCorp, although IBM Cloud does maintain their own Terraform provider.

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

NEW QUESTION 82

- (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 85

- (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 88

- (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 93

- (Exam Topic 2)

Which of the following Terraform files should be ignored by Git when committing code to a repo? (select Three)

- A. Files named exactly terraform.tfvars or terraform.tfvars.json.
- B. Any files with names ending in .auto.tfvars or .auto.tfvars.json.
- C. input.tf
- D. terraform.tfstate
- E. output.tf

Answer: ABD

Explanation:

The .gitignore file should be configured to ignore Terraform files that either contain sensitive data or are not required to save. Terraform state (terraform.tfstate) can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords. When using local state, state is stored in plain-text JSON files. The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others.

NEW QUESTION 95

- (Exam Topic 2)

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

- A. False
- B. True

Answer: A

Explanation:

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

NEW QUESTION 100

- (Exam Topic 2)

You have declared a variable name my_var in terraform configuration without a value associated with it. variable my_var {}
After running terraform plan it will show an error as variable is not defined.

- A. True
- B. False

Answer: B

Explanation:

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

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

```
variable "variable_name" {}  
terraform apply -var variable_name="value"
```

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

String

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

```
variable "template" { type = string  
default = "01000000-0000-4000-8000-000030080200"  
}
```

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

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

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

```
variable "users" { type = list  
default = ["root", "user1", "user2"]  
}
```

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

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

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

```
variable "plans" { type = map default = {  
"5USD" = "1xCPU-1GB" "10USD" = "1xCPU-2GB" "20USD" = "2xCPU-4GB"  
}
```

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

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

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

```
variable "storage_sizes" { type = map  
default = {  
"1xCPU-1GB" = "25"  
"1xCPU-2GB" = "50"  
"2xCPU-4GB" = "80"  
}
```

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

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

Boolean

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

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

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

```
create_password = var.set_password
```

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

```
terraform apply -var set_password="true"
```

NEW QUESTION 105

- (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 106

- (Exam Topic 2)

Refer to the below code where developer is outputting the value of the database password but has used sensitive parameter to hide the output value in the CLI.
output "db_password" { value = aws_db_instance.db.password description = "The password for logging in to the database." sensitive = true}
Since sensitive is set to true, the value associated with db password will not be present in state file as plain-text?

- A. False
- B. True

Answer: A

Explanation:

Sensitive output values are still recorded in the state, and so will be visible to anyone who is able to access the state data.

NEW QUESTION 108

- (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 113

- (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 115

- (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 118

- (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 121

- (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 122

- (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 126

- (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 131

- (Exam Topic 3)

In regards to Terraform state file, select all the statements below which are correct?

- A. When using local state, the state file is stored in plain-text.

- B. The state file is always encrypted at rest.
- C. Storing state remotely can provide better security.
- D. Using the mask feature, you can instruct Terraform to mask sensitive data in the state file.
- E. The Terraform state can contain sensitive data, therefore the state file should be protected from unauthorized access.
- F. Terraform Cloud always encrypts state at rest.

Answer: ACEF

Explanation:

Terraform state can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.

When using local state, state is stored in plain-text JSON files.

When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.

Storing Terraform state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

Recommendations

If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.

Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

For example:

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

* The S3 backend supports encryption at rest when the encrypt option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

NEW QUESTION 132

- (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 133

- (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 136

- (Exam Topic 3)

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

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

Answer: D

Explanation:

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

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

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

NEW QUESTION 137

- (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 141

- (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 142

- (Exam Topic 3)

Which of the following variable definition files will terraform load automatically?

- A. terraform.tfvar
- B. Any files with names ending in .auto.tfvars.json
- C. terraform.tfvars
- D. terraform.tfvars.json

Answer: BCD

Explanation:

Terraform also automatically loads a number of variable definitions files if they are present: Files named exactly terraform.tfvars or terraform.tfvars.json.

Any files with names ending in .auto.tfvars or .auto.tfvars.json. <https://www.terraform.io/docs/configuration/variables.html>

<https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files>

NEW QUESTION 146

- (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 150

- (Exam Topic 3)

You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.

- A. Run the command export TF_LOG_FILE=./terraform.log.
- B. Run the command export TF_LOG_PATH=./terraform.log.
- C. Run the command export TF_DEBUG_PATH=./terraform.log.
- D. No explicit action require
- E. Terraform will take care of this as you have enable TF_LOG.

Answer: B

Explanation:

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

NEW QUESTION 152

- (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 154

- (Exam Topic 3)

What kind of resource dependency is stored in terraform.tfstate file?

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

Answer: A

Explanation:

Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state.

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

NEW QUESTION 159

- (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 161

- (Exam Topic 3)

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

- A. False
- B. True

Answer: A

Explanation:

If you need to migrate to another backend, such as Terraform Cloud, so you can continue managing it. By migrating your Terraform state, you can hand off infrastructure without de-provisioning anything.

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

NEW QUESTION 162

- (Exam Topic 3)

By default, provisioners that fail will also cause the Terraform apply itself to error. How can you change this default behavior within a provisioner?

- A. provisioner "local-exec" { on_failure = "next" }
- B. provisioner "local-exec" { when = "failure" terraform apply }
- C. provisioner "local-exec" { on_failure = "continue" }
- D. provisioner "local-exec" { on_failure = continue }

Answer: C

Explanation:

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

NEW QUESTION 165

- (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 169

- (Exam Topic 3)

A single terraform resource file that defines an aws_instance resource can simply be renamed to vsphere_virtual_machine in order to switch cloud providers.

A. True

B. False

Answer: B

Explanation:

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

NEW QUESTION 172

- (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 176

- (Exam Topic 3)

Which of the following are string functions? Select three

A. tostring

B. tonumber

C. Chomp

D. format

E. join

Answer: CDE

Explanation:

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

NEW QUESTION 178

- (Exam Topic 3)

What does terraform refresh command do?

A. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.

B. terraform refresh command basically updates the configuration file with the current state of the actual infrastructure

C. terraform refresh is use to change/modify the infrastructure based on the existing state file, at that moment.

D. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.

E. terraform refresh syncs the state file with the real world infrastructure.

Answer: E

NEW QUESTION 181

- (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 183

- (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 185

- (Exam Topic 4)

terraform init retrieves the source code tot all referenced modules

- A. True
- B. False

Answer: A

Explanation:

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

NEW QUESTION 189

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 190

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 195

- (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 198

- (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 199

- (Exam Topic 4)

What is the result of the following terraform function call?

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 200

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 203

- (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. terreform providers- upgrade
- B. terreform apply -upgrade
- C. terreform refresh -upgrade
- D. terreformn Init -upgrade

Answer: D

NEW QUESTION 208

- (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 210

- (Exam Topic 4)

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

- A. Initializes downloaded and/or installed providers
- B. Initializes the backend configuration
- C. Provisions the declared resources in your configuration

D. Download the declared providers which are supported by HashiCorp

Answer: ABD

Explanation:

The terraform init command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times. This command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing configuration or state. terraform init command does * Copy a Source Module
* Backend Initialization
* Child Module Installation
* Plugin Installation <https://www.terraform.io/docs/commands/init.html>

NEW QUESTION 214

- (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 219

- (Exam Topic 4)

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

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

Answer: BC

Explanation:

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

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

NEW QUESTION 221

- (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 223

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 227

- (Exam Topic 4)

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 231

- (Exam Topic 4)

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

Original configuration:

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

Updated configuration:

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

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

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

Answer: C

Explanation:

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

NEW QUESTION 234

- (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 238

- (Exam Topic 4)

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

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

Answer: CD

Explanation:

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

NEW QUESTION 240

- (Exam Topic 4)

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

- A. True
- B. False

Answer: A

NEW QUESTION 242

- (Exam Topic 4)

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

```
* 1. resource "aws_route53_record" "www"
* 2. {
* 3.   zone_id = aws_route53_zone.primary.zone_id
* 4.   name = "www.example.com"
* 5.   type = "A"
* 6.   ttl = "300"
* 7.   records = [module.web_server.instance_ip_address] 8. }
```

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

Answer: B

Explanation:

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

NEW QUESTION 246

- (Exam Topic 4)

Which statements best describes what the local variable assignment is doing in the following code snippet:

- A. Create a distinct list of route table name objects
- B. Create a map of route table names to subnet names
- C. Create a map of route table names from a list of subnet names
- D. Create a list of route table names eliminating duplicates

Answer: D

NEW QUESTION 251

- (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 256

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 260

- (Exam Topic 4)

A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?

- A. terraform workspace dev
- B. terraform workspace select dev
- C. terraform workspace -switch dev
- D. terraform workspace switch dev

Answer: B

Explanation:

The terraform workspace select command is used to choose a different workspace to use for further operations.

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

NEW QUESTION 263

- (Exam Topic 4)

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

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

Answer: B

Explanation:

Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state. <https://www.terraform.io/docs/state/purpose.html#metadata>

NEW QUESTION 267

- (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 268

- (Exam Topic 4)

terraform destroy is the only way to remove infrastructure.

- A. True
- B. False

Answer: B

NEW QUESTION 270

- (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 272

- (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 273

- (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 276

- (Exam Topic 4)

Do terraform workspaces help in adding/allowing multiple state files for a single configuration?

- A. True
- B. False

Answer: A

NEW QUESTION 278

- (Exam Topic 4)

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

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

  ebs_block_device {
    device_name = "sda2"
    volume_size = 16
  }

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

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

Answer: A

Explanation:

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

NEW QUESTION 283

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 286

- (Exam Topic 4)

What does terraform import allow you to do?

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

Answer: C

NEW QUESTION 291

- (Exam Topic 4)

Your company has a lot of workloads in AWS , and Azure that were respectively created using CloudFormation , and AzureRM Templates. However , now your CIO has decided to use Terraform for all new projects , and has asked you to check how to integrate the existing environment with terraform code. What should be your next plan of action?

- A. Tell the CIO that this is not possible . Resources created in CloudFormation , and AzureRM templates cannot be tracked using terraform.
- B. Use terraform import command to import each resource one by one .
- C. This is only possible in Terraform Enterprise , which has the TerraformConverter exe that can take any other template language like AzureRM and convert to Terraform code.
- D. Just write the terraform config file for the new resources , and run terraform apply , the state file will automatically be updated with the details of the new resources to be imported.

Answer: B

NEW QUESTION 292

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 297

- (Exam Topic 4)

A terraform apply can not _____ infrastructure.

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

Answer: A

NEW QUESTION 300

- (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 302

- (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 304

- (Exam Topic 4)

Select two answers to complete the following sentence: Before a new provider can be used, it must be _____ and _____.

- A. approved by HashiCorp
- B. uploaded to source control
- C. declared in the configuration
- D. initialized

Answer: CD

Explanation:

Each time a new provider is added to configuration -- either explicitly via a provider block or by adding a resource from that provider -- Terraform must initialize the provider before it can be used. Initialization downloads and installs the provider's plugin so that it can later be executed.

NEW QUESTION 307

- (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 309

- (Exam Topic 4)

The following is a snippet from a Terraform configuration file: Which, when validated, results in the following error: Fill in the blank in the error message with the correct string from the list below.

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

Answer: D

Explanation:

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

NEW QUESTION 312

- (Exam Topic 4)

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

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

Answer: C

Explanation:

If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply .

NEW QUESTION 315

- (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 319

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 324

- (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 326

- (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 331

- (Exam Topic 4)

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

- A. TF_help
- B. TF LOG
- C. TF_Debug
- D. TF_var_log

Answer: B

NEW QUESTION 333

- (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. var!

Answer: B

NEW QUESTION 335

- (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 340

- (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 342

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 343

- (Exam Topic 4)

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

- A. True
- B. False

Answer: A

Explanation:

Organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g.

staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are not a suitable isolation mechanism for this scenario.
<https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces>

NEW QUESTION 347

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 348

- (Exam Topic 4)

Which of the following is considered a Terraform plugin?

- A. Terraform language
- B. Terraform tooling
- C. Terraform logic
- D. Terraform provider

Answer: D

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.

<https://www.terraform.io/docs/plugins/basics.html>

NEW QUESTION 352

- (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 353

- (Exam Topic 4)

Changing the Terraform backend from the default "local" backend to a different one after doing your first terraform apply is:

- A. Mandatory
- B. Optional
- C. Impossible
- D. Discouraged

Answer: B

NEW QUESTION 357

- (Exam Topic 4)

A "backend" in Terraform determines how state is loaded and how an operation such as apply is executed. Which of the following is not a supported backend type?

- A. Terraform enterprise
- B. Consul
- C. Github

- D. S3
- E. Artifactory

Answer: C

Explanation:

Github is not a supported backend type. <https://www.terraform.io/docs/backends/types/index.html>

NEW QUESTION 360

- (Exam Topic 4)

After executing a terraform apply, you notice that a resource has a tilde (~) next to it. What does this infer?

- A. The resource will be updated in place.
- B. The resource will be created.
- C. Terraform can't determine how to proceed due to a problem with the state file.
- D. The resource will be destroyed and recreated.

Answer: A

Explanation:

The prefix -/+ means that Terraform will destroy and recreate the resource, rather than updating it in-place. The prefix ~ means that some attributes and resources can be updated in-place.

\$ terraform apply

aws_instance.example: Refreshing state... [id=i-0bbf06244e44211d1] An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

-/+ destroy and then create replacement Terraform will perform the following actions:

aws_instance.example must be replaced

-/+ resource "aws_instance" "example" {

~ ami = "ami-2757f631" -> "ami-b374d5a5" # forces replacement

~ arn = "arn:aws:ec2:us-east-1:130490850807:instance/i-0bbf06244e44211d1" -> (known after apply)

~ associate_public_ip_address = true -> (known after apply)

~ availability_zone = "us-east-1c" -> (known after apply)

~ cpu_core_count = 1 -> (known after apply)

~ cpu_threads_per_core = 1 -> (known after apply)

- disable_api_termination = false -> null

- ebs_optimized = false -> null get_password_data = false

+ host_id = (known after apply)

~ id = "i-0bbf06244e44211d1" -> (known after apply)

~ instance_state = "running" -> (known after apply) instance_type = "t2.micro"

~ ipv6_address_count = 0 -> (known after apply)

~ ipv6_addresses = [] -> (known after apply)

+ key_name = (known after apply)

- monitoring = false -> null

+ network_interface_id = (known after apply)

+ password_data = (known after apply)

+ placement_group = (known after apply)

~ primary_network_interface_id = "eni-0f1ce5bdae258b015" -> (known after apply)

~ private_dns = "ip-172-31-61-141.ec2.internal" -> (known after apply)

~ private_ip = "172.31.61.141" -> (known after apply)

~ public_dns = "ec2-54-166-19-244.compute-1.amazonaws.com" -> (known after apply)

~ public_ip = "54.166.19.244" -> (known after apply)

~ security_groups = [

- "default",

] -> (known after apply) source_dest_check = true

~ subnet_id = "subnet-1facdf35" -> (known after apply)

~ tenancy = "default" -> (known after apply)

~ volume_tags = {} -> (known after apply)

~ vpc_security_group_ids = [

- "sg-5255f429",

] -> (known after apply)

- credit_specification {

- cpu_credits = "standard" -> null

}

+ ebs_block_device {

+ delete_on_termination = (known after apply)

+ device_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ snapshot_id = (known after apply)

+ volume_id = (known after apply)

+ volume_size = (known after apply)

+ volume_type = (known after apply)

}

+ ephemeral_block_device {

+ device_name = (known after apply)

+ no_device = (known after apply)

+ virtual_name = (known after apply)

}

+ network_interface {

+ delete_on_termination = (known after apply)

+ device_index = (known after apply)

+ network_interface_id = (known after apply)

```

}
~ root_block_device {
~ delete_on_termination = true -> (known after apply)
~ iops = 100 -> (known after apply)
~ volume_id = "vol-0079e485d9e28a8e5" -> (known after apply)
~ volume_size = 8 -> (known after apply)
~ volume_type = "gp2" -> (known after apply)
}
}
}

```

Plan: 1 to add, 0 to change, 1 to destroy.

NEW QUESTION 363

- (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 364

- (Exam Topic 4)

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

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

Answer: D

NEW QUESTION 367

- (Exam Topic 4)

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 370

- (Exam Topic 4)

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

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

Answer: ADE

NEW QUESTION 374

- (Exam Topic 4)

```
resource "aws_s3_bucket" "example" { bucket = "my-test-s3-terraform-bucket" ...} resource "aws_iam_role" "test_role" { name = "test_role" ...}
```

Due to the way that the application code is written, the s3 bucket must be created before the test role is created, otherwise there will be a problem. How can you ensure that?

- A. This will already be taken care of by terraform native implicit dependenc
- B. Nothing else needs to be done from your end.
- C. Add explicit dependency using depends_on. This will ensure the correct order of resource creation.

- D. Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.
 E. This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.

Answer: B

Explanation:

Use the depends_on meta-argument to handle hidden resource dependencies that Terraform can't automatically infer. Explicitly specifying a dependency is only necessary when a resource relies on some other resource's behavior but doesn't access any of that resource's data in its arguments.

NEW QUESTION 379

- (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 381

- (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 382

- (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 384

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

NEW QUESTION 388

- (Exam Topic 4)

What does state locking accomplish?

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

Answer: C

Explanation:

If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state. Source: <https://www.terraform.io/language/state/locking>

NEW QUESTION 389

- (Exam Topic 4)

Which is the best way to specify a tag of v1.0.0 when referencing a module stored in Git (for example `git::https://example.com/vpc.git`)?

- A. Append `ref=v1.0.0` argument to the source path Most Voted
- B. Add `version = "1.0.0"` parameter to module block
- C. Nothing "modules stored on GitHub always default to version 1.0.0"
- D. Modules stored on GitHub do not support versioning

Answer: A

Explanation:

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

NEW QUESTION 394

- (Exam Topic 4)

Given the Terraform configuration below, in which order will the resources be created?

```
* 1. resource "aws_instance" "web_server"
* 2. {
* 3. ami = "ami-b374d5a5"
* 4. instance_type = "t2.micro"
* 5. }
* 6. resource "aws_eip" "web_server_ip"
* 7. {
* 8. vpc = true instance = aws_instance.web_server.id
* 9. }
```

- A. `aws_eip` will be created first `aws_instance` will be created second
- B. `aws_eip` will be created first `aws_instance` will be created second
- C. Resources will be created simultaneously
- D. `aws_instance` will be created first `aws_eip` will be created second

Answer: D

Explanation:

Implicit and Explicit Dependencies

By studying the resource attributes used in interpolation expressions, Terraform can automatically infer when one resource depends on another. In the example above, the reference to `aws_instance.web_server.id` creates an implicit dependency on the `aws_instance` named `web_server`.

Terraform uses this dependency information to determine the correct order in which to create the different resources.

```
# Example of Implicit Dependency resource "aws_instance" "web_server" { ami = "ami-b374d5a5"
instance_type = "t2.micro"
}
```

```
resource "aws_eip" "web_server_ip" { vpc = true
instance = aws_instance.web_server.id
}
```

In the example above, Terraform knows that the `aws_instance` must be created before the `aws_eip`. Implicit dependencies via interpolation expressions are the primary way to inform Terraform about these relationships, and should be used whenever possible.

Sometimes there are dependencies between resources that are not visible to Terraform. The `depends_on` argument is accepted by any resource and accepts a list of resources to create explicit dependencies for.

For example, perhaps an application we will run on our EC2 instance expects to use a specific Amazon S3 bucket, but that dependency is configured inside the application code and thus not visible to Terraform. In that case, we can use `depends_on` to explicitly declare the dependency:

```
# Example of Explicit Dependency
```

```
# New resource for the S3 bucket our application will use. resource "aws_s3_bucket" "example" {
bucket = "terraform-getting-started-guide" acl = "private"
}
```

```
# Change the aws_instance we declared earlier to now include "depends_on" resource "aws_instance" "example" {
ami = "ami-2757f631" instance_type = "t2.micro"
```

```
# Tells Terraform that this EC2 instance must be created only after the
# S3 bucket has been created. depends_on = [aws_s3_bucket.example]
}
```

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

NEW QUESTION 399

- (Exam Topic 4)
Terraform Cloud is available only as a paid offering from HashiCorp.

- A. True
- B. False

Answer: B

Explanation:

Many of Terraform Cloud features are free for small teams, including remote state storage, remote runs, and VCS connections. "Terraform Cloud is a commercial SaaS product developed by HashiCorp. Many of its features are free for small teams, including remote state storage, remote runs, and VCS connections. We also offer paid plans for larger teams that include additional collaboration and governance features."

NEW QUESTION 403

- (Exam Topic 4)
What does Terraform use .terraform.lock.hcl file for?

- A. Tracking provider dependencies Most Voted
- B. There is no such file
- C. Preventing Terraform runs from occurring
- D. Storing references to workspaces which are locked

Answer: A

Explanation:

<https://www.terraform.io/language/files/dependency-lock>
"hcl , and this name is intended to signify that it is a lock file for various items that Terraform caches in the . terraform subdirectory of your working directory. Terraform automatically creates or updates the dependency lock file each time you run the terraform init command."

NEW QUESTION 406

- (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 408

- (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 411

- (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 415

- (Exam Topic 4)

Select all features which are exclusive to Terraform Enterprise. (Select Three)

- A. Sentinel
- B. Cost Estimation
- C. Audit Logs
- D. Clustering
- E. SAML/SSO

Answer: CDE

Explanation:

Sentinel and Cost Estimation are also available in Terraform Cloud <https://www.hashicorp.com/products/terraform/pricing/>

NEW QUESTION 419

- (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 422

- (Exam Topic 4)

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

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

Answer: D

NEW QUESTION 426

- (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 431

- (Exam Topic 4)

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

- A. Passing ---destroy at the end of api 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 433

- (Exam Topic 4)

Given the Terraform configuration below, in which order will the resources be created?

- A. Larger image

- B. resources will be created simultaneously
- C. aws_eip will be created first aws_instance will be created second
- D. aws_instance will be created first aws_eip will be created second

Answer: D

Explanation:

The aws_instance will be created first, and then aws_eip will be created second due to the aws_eip's resource dependency of the aws_instance id

NEW QUESTION 436

- (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 440

- (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 445

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questions and Answers in PDF Format

TA-002-P Practice Exam Features:

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

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