HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1
- (Exam Topic 1)
Which of the following is available only in Terraform Enterprise or Cloud workspaces and not in Terraform CLI?

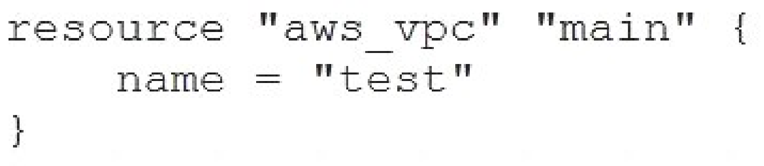
A. Secure variable storage
B. Support for multiple cloud providers
C. Dry runs with terraform plan
D. Using the workspace as a data source

Answer: A

Explanation:

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

NEW QUESTION 2
- (Exam Topic 1)
What is the provider for this fictitious resource?



```
resource "aws_vpc" "main" {
    name = "test"
}
```

A. vpc
B. main
C. aws
D. test

Answer: C

Explanation:

Reference: https://docs.aws.amazon.com/cloudformation-cli/latest/userguide/resource-types.html

NEW QUESTION 3
- (Exam Topic 1)
When you initialize Terraform, where does it cache modules from the public Terraform Module Registry?

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

Answer: C

Explanation:

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

NEW QUESTION 4
- (Exam Topic 1)
Terraform provisioners can be added to any resource block.

A. True
B. False

Answer: A

Explanation:
https://www.phillipsj.net/posts/introduction-to-terraform-provisioners/
As you continue learning about Terraform, you will start hearing about provisioners. Terraform provisioners can be created on any resource and provide a way to execute actions on local or remote machines.
https://www.terraform.io/language/resources/provisioners/local-exec

NEW QUESTION 5
- (Exam Topic 1)
Terraform can only manage resource dependencies if you set them explicitly with the depends_on argument.

A. True
B. False

Answer: A

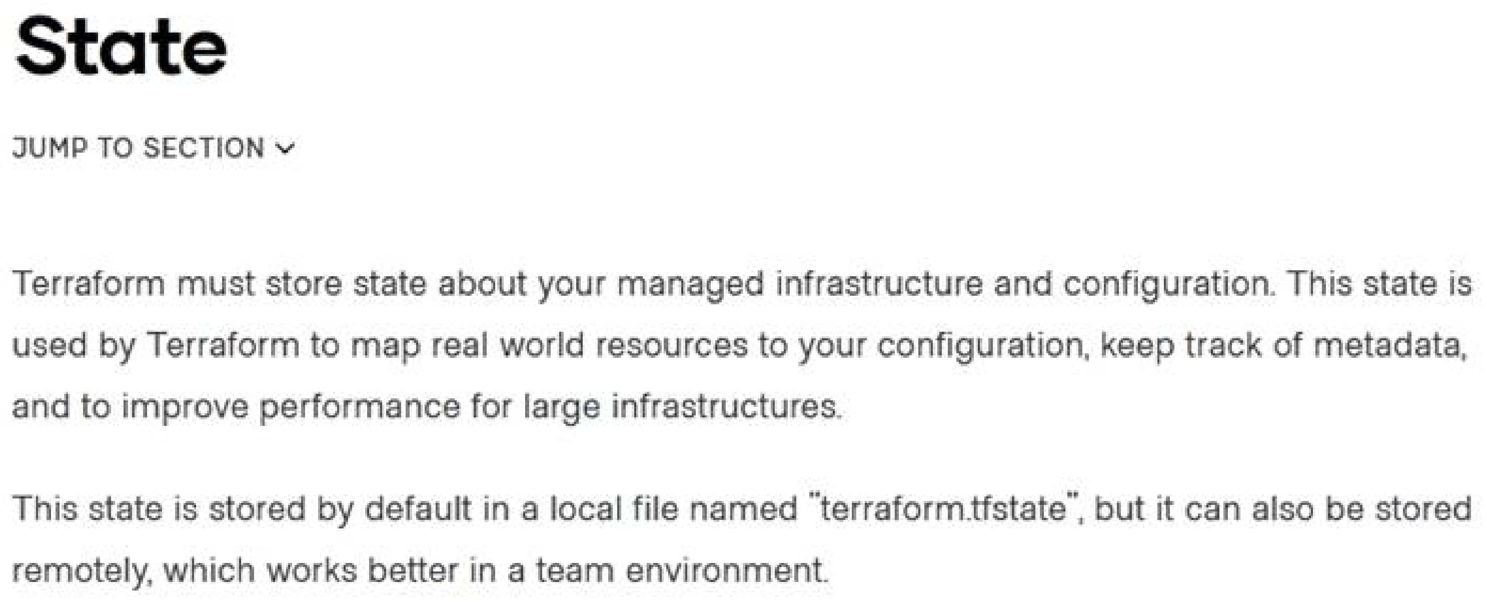
Explanation:
"Use the depends_on meta-argument to handle hidden resource or module dependencies that Terraform cannot automatically infer. You only need to explicitly specify a dependency when a resource or module relies on another resource's behavior but does not access any of that resource's data in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

NEW QUESTION 6
- (Exam Topic 1)
FILL BLANK
What is the name of the default file where Terraform stores the state?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
"This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment."
https://www.terraform.io/language/state



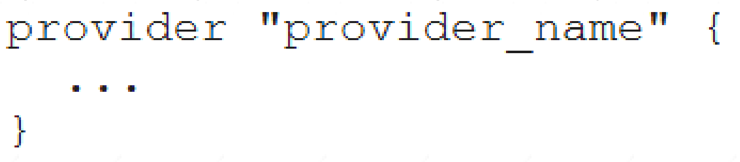
State

JUMP TO SECTION

Terraform must store state about your managed infrastructure and configuration. This state is used by Terraform to map real world resources to your configuration, keep track of metadata, and to improve performance for large infrastructures.

This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

NEW QUESTION 7
- (Exam Topic 1)
A provider configuration block is required in every Terraform configuration. Example:



```
provider "provider_name" {
  ...
}
```

A. True
B. False

Answer: B

Explanation:
Unlike many other objects in the Terraform language, a provider block may be omitted if its contents would otherwise be empty. Terraform assumes an empty default configuration for any provider that is not explicitly configured. https://www.terraform.io/language/providers/configuration

NEW QUESTION 8
- (Exam Topic 1)
A terraform apply can not ______ infrastructure.

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

Answer: D

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

NEW QUESTION 9
- (Exam Topic 1)
Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/downloads

NEW QUESTION 10
- (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 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 13
- (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 16
- (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 18
- (Exam Topic 1)
What type of block is used to construct a collection of nested configuration blocks?

A. for_each
B. repeated
C. nesting
D. dynamic

Answer: D

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

NEW QUESTION 19
- (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 23
- (Exam Topic 1)
When does terraform apply reflect changes in the cloud environment?

A. Immediately
B. However long it takes the resource provider to fulfill the request
C. After updating the state file
D. Based on the value provided to the -refresh command line argument
E. None of the above

Answer: B

NEW QUESTION 26
- (Exam Topic 1)
HashiCorp Configuration Language (HCL) supports user-defined functions.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/language/functions
The Terraform language does not support user-defined functions, and so only the functions built into the language are available for use

NEW QUESTION 29
- (Exam Topic 1)
Terraform and Terraform providers must use the same major version number in a single configuration.

A. True
B. False

Answer: B

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

NEW QUESTION 30
- (Exam Topic 1)
What does the default "local" Terraform backend store?

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

Answer: D

Explanation:
The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.
Reference: https://www.terraform.io/docs/language/settings/backends/local.html

NEW QUESTION 31
- (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 36
- (Exam Topic 1)
If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?

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

Answer: A

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

NEW QUESTION 39
- (Exam Topic 2)
What is the standard workflow that a developer follows while working with terraform open source version?

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

Answer: D

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

NEW QUESTION 40
- (Exam Topic 2)
Environment variables can be used to set variables. The environment variables must be in the format " "_<variablename>. Select the correct prefix string from the following list.

A. TF_CLI_ARGS
B. TF_VAR
C. TF_VAR_
D. TF_VAR_ENV

Answer: C

Explanation:
Environment variables can be used to set variables. The environment variables must be in the format TF_VAR_name and this will be checked last for a value. For example:
export TF_VAR_region=us-west-1
export TF_VAR_ami=ami-049d8641 export TF_VAR_alist='[1,2,3]'
export TF_VAR_amap='{ foo = "bar", baz = "qux" }' https://www.terraform.io/docs/commands/environment-variables.html

NEW QUESTION 43
- (Exam Topic 2)
Which of the following represents a feature of Terraform Cloud that is NOT free to customers?

A. Roles and Team Management
B. WorkSpace Management
C. Private Module Registry
D. VCS Integration

Answer: A

Explanation:
Role Based Access Controls (RBAC) for controlling permissions for who has access to what configurations within an organization and it is not free to customers.
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 44
- (Exam Topic 2)
In regards to deploying resources in multi-cloud environments, what are some of the benefits of using Terraform rather than a provider's native tooling? (select three)

A. Terraform can help businesses deploy applications on multiple clouds and on-premises infrastructure.
B. Terraform is not cloud-agnostic and can be used to deploy resources across a single public cloud.
C. Terraform simplifies management and orchestration, helping operators build large-scale, multi-cloud infrastructure.
D. Terraform can manage cross-cloud dependencies.

Answer: ACD

Explanation:
Terraform is cloud-agnostic and allows a single configuration to be used to manage multiple providers, and to even handle cross-cloud dependencies. This simplifies management and orchestration, helping operators build large-scale multi-cloud infrastructures.
https://www.terraform.io/intro/use-cases.html

NEW QUESTION 49
- (Exam Topic 2)
How can you ensure that the engineering team who has access to git repo will not create any non-compliant resources that might lead to a security audit failure in future. your team is using Hashicorp Terraform Enterprise Edition.

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

Answer: C

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

NEW QUESTION 52
- (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 54
- (Exam Topic 2)
Terraform import command can import resources into modules as well directly into the root of your state.

A. True
B. False

Answer: A

Explanation:
Import will find the existing resource from ID and import it into your Terraform state at the given ADDRESS. ADDRESS must be a valid resource address. Because any resource address is valid, the import command can import resources into modules as well directly into the root of your state.
Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.
This is a great way to slowly transition infrastructure to Terraform.
The terraform import command is used to import existing infrastructure.
To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform. For example:

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

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

```
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.
```

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.
As a result of the above command, the resource is recorded in the state file. We can now run terraform plan to see how the configuration compares to the imported resource, and make any adjustments to the configuration to align with the current (or desired) state of the imported object.
https://www.terraform.io/docs/commands/import.html

NEW QUESTION 57
- (Exam Topic 2)
Which of the following best describes a Terraform provider?

A. A plugin that Terraform uses to translate the API interactions with the service or provider.
B. Serves as a parameter for a Terraform module that allows a module to be customized.
C. Describes an infrastructure object, such as a virtual network, compute instance, or other components.
D. A container for multiple resources that are used together.

Answer: A

Explanation:
A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g.Terraform Cloud, DNSimple, Cloudflare).
https://www.terraform.io/docs/providers/index.html

NEW QUESTION 59
- (Exam Topic 2)
Which of the following best describes the default local backend?

A. The local backend is where Terraform Enterprise stores logs to be processed by an log collector.
B. The local backend stores state on the local filesystem, locks the state using system APIs, and performs operations locally.
C. The local backend is the directory where resources deployed by Terraform have direct access to in order to update their current state.
D. The local backend is how Terraform connects to public cloud services, such as AWS, Azure, or GCP.

Answer: B

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

```
terraform { backend "local" {
path = "relative/path/to/terraform.tfstate"
}
}
```

https://www.terraform.io/docs/backends/types/local.html

NEW QUESTION 60
- (Exam Topic 2)
What is the purpose of using the local-exec provisioner? (Select Two)

A. To invoke a local executable.
B. Executes a command on the resource to invoke an update to the Terraform state.
C. To execute one or more commands on the machine running Terraform.
D. Ensures that the resource is only executed in the local infrastructure where Terraform is deployed.

Answer: AC

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. Note that even though the resource will be fully created when the provisioner is run, there is no guarantee that it will be in an operable state - for example system services such as sshd may not be started yet on compute resources.
Example usage

```
resource "aws_instance" "web" {
# ...
provisioner "local-exec" {
command = "echo ${aws_instance.web.private_ip} >> private_ips.txt"
}
}
```

Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives.
https://www.terraform.io/docs/provisioners/local-exec.html

NEW QUESTION 63
- (Exam Topic 3)
A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip"} What is the best way to ensure that you are not using the latest version of the module?

A. Add a module version constraint in your configuration's backend block and specify a previous version.
B. Add a version key to the module configuration and specify a previous version.
C. Delete the latest version of the module in S3 to rollback to the previous version.
D. Add a version property to the module in Terraform's state file and specify a previous version.

Answer: C

Explanation:
Version constraints are supported only for modules installed from a module registry, such as the Terraform Registry or Terraform Cloud's private module registry. Other module sources can provide their own versioning mechanisms within the source string itself, or might not support versions at all. In particular, modules sourced from local file paths do not support version; since they're loaded from the same source repository.
Only Terraform Registries support module versioning by using the version key, one cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version.
https://www.terraform.io/docs/configuration/modules.html#source

NEW QUESTION 64
- (Exam Topic 3)
Which of the following state management command allow you to retrieve a list of resources that are part of the state file?

A. terraform state list
B. terraform state view

C. terraform view
D. terraform list

Answer: A

Explanation:
The terraform state list command is used to list resources within a Terraform state. Usage: terraform state list [options] [address...]
The command will list all resources in the state file matching the given addresses (if any). If no addresses are given, all resources are listed.
https://www.terraform.io/docs/commands/state/list.html

NEW QUESTION 66
- (Exam Topic 3)
You have created two workspaces PROD and DEV. You have switched to DEV and provisioned DEV infrastructure from this workspace. Where is your state file stored?

A. terraform.d
B. terraform.tfstate
C. terraform.tfstate.DEV
D. terraform.tfstate.d

Answer: D

Explanation:
Terraform stores the workspace states in a directory called terraform.tfstate.d. This directory should be treated similarly to default workspace state file
terraform.tfstate main.tf
provider.tf terraform.tfstate.d DEV
terraform.tfstate # DEV workspace state file PROD
terraform.tfstate # PROD workspace state file terraform.tfvars # Default workspace state file variables.tf

NEW QUESTION 68
- (Exam Topic 3)
The canonical format may change in minor ways between Terraform versions, so after upgrading Terraform it is recommended to proactively run.

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

Answer: A

NEW QUESTION 70
- (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 71
- (Exam Topic 3)
After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?

A. ERROR
B. INFO
C. WARN
D. TRACE
E. DEBUG

Answer: D

Explanation:
Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr.

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

NEW QUESTION 72
- (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 73
- (Exam Topic 3)
The Security Operations team of ABC Enterprise wants to mandate that all the Terraform configuration that creates an S3 bucket must have encryption feature enabled. What is the best way to achieve it?

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

Answer: A

Explanation:
Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.
Using Sentinel with Terraform Cloud involves:
* Defining the policies - Policies are defined using the policy language with imports for parsing the Terraform plan, state and configuration.
* Managing policies for organizations - Users with permission to manage policies can add policies to their organization by configuring VCS integration or uploading policy sets through the API. They also define which workspaces the policy sets are checked against during runs. (More about permissions.)
* Enforcing policy checks on runs - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is executed.
* Mocking Sentinel Terraform data - Terraform Cloud provides the ability to generate mock data for any run within a workspace. This data can be used with the Sentinel CLI to test policies before deployment.
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION 76
- (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 81
- (Exam Topic 3)
Complete the following sentence:
For local state, the workspaces are stored directly in a ______.

A. a file called terraform.tfstate.backup
B. directory called terraform.workspaces.tfstate

C. a file called terraform.tfstate
D. directory called terraform.tfstate.d

Answer: D

Explanation:
For local state, Terraform stores the workspace states in a directory called terraform.tfstate.d. https://www.terraform.io/docs/state/workspaces.html#workspace-internals

NEW QUESTION 83
- (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 86
- (Exam Topic 3)
Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?

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

Answer: A

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

NEW QUESTION 87
- (Exam Topic 3)
You have created a terraform script that uses a lot of new constructs that have been introduced in terraform v0.12. However, many developers who are cloning the script from your git repo, are using v0.11, and getting errors. What can be done from your end to solve this problem?

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

Answer: A

Explanation:
https://www.terraform.io/docs/configuration/terraform.html

NEW QUESTION 89
- (Exam Topic 3)
Multiple configurations for the same provider can be used in a single configuration file.

A. False
B. True

Answer: B

Explanation:
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

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

The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances

NEW QUESTION 94
- (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 98
- (Exam Topic 4)
A Terraform output that sets the "sensitive" argument to true will not store that value in the state file.

A. True
B. False

Answer: B

Explanation:
Reference: https://www.terraform.io/language/values/outputs

NEW QUESTION 99
- (Exam Topic 4)
Valarie has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code. Valarie wants to hide the output value in the CLI after terraform apply that's why she has used sensitive parameter.
* 1. output "db_password" {
* 2. value = local.db_password
* 3. sensitive = true
* 4. }
Since sensitive is set to true, will the value associated with db password be available in plain-text in the state file for everyone to read?

A. Yes
B. No

Answer: A

Explanation:
Outputs can be marked as containing sensitive material by setting the sensitive attribute to true, like this: output "sensitive" {
sensitive = true value = VALUE
}
When outputs are displayed on-screen following a terraform apply or terraform refresh, sensitive outputs are redacted, with <sensitive> displayed in place of their value.
Limitations of Sensitive Outputs
The values of sensitive outputs are still stored in the Terraform state, and available using the terraform output command, so cannot be relied on as a sole means of protecting values.
Sensitivity is not tracked internally, so if the output is interpolated in another module into a resource, the value will be displayed.

NEW QUESTION 104
- (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 108
- (Exam Topic 4)
What is the purpose of a Terraform workspace in either open source of enterprise?

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

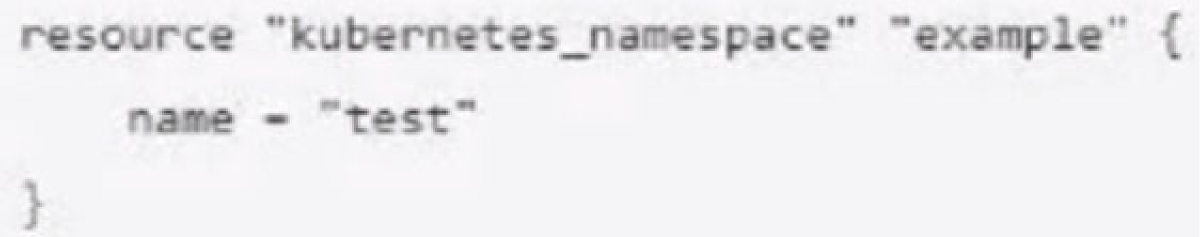
Answer: B

NEW QUESTION 111
- (Exam Topic 4)
Terraform is currently being used by your organisation to create resources on AWS for the development of a web application. One of your coworkers wants to change the instance type to "t2.large" while keeping the default set values.
What adjustments does the teammate make in order to meet his goal?

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

Answer: D

NEW QUESTION 115
- (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 117
- (Exam Topic 4)
True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.

A. False
B. True

Answer: A

Explanation:

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

NEW QUESTION 122
- (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 127
- (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 132
- (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 137
- (Exam Topic 4)
Your risk management organization requires that new AWS S3 buckets must be private and encrypted at rest. How can Terraform Enterprise automatically and proactively enforce this security control?

A. With a Sentinel policy, which runs before every apply
B. By adding variables to each TFE workspace to ensure these settings are always enabled
C. With an S3 module with proper settings for buckets
D. Auditing cloud storage buckets with a vulnerability scanning tool

Answer: A

Explanation:
https://docs.hashicorp.com/sentinel/intro/what
https://medium.com/hashicorp-engineering/enforcing-aws-s3-security-best-practice-using-terraform-sentinel-dd

NEW QUESTION 142
- (Exam Topic 4)
Terraform console provides an interactive command-line console for evaluating and experimenting with expressions. You can use it to test interpolations before using them in configurations and to interact with any values currently saved in state.
Which configuration consistency errors does terraform validate report?

A. A mix of spaces and tabs in configuration files
B. Differences between local and remote state
C. Terraform module isn't the latest version
D. Declaring a resource identifier more than once

Answer: D

Explanation:
validate will look for syntax errors "Declaring a resource identifier more than once" is a syntax error

NEW QUESTION 143
- (Exam Topic 4)
John is writing a module and within the module, there are multiple places where he has to use the same
conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration,. What is a better approach to dealing with this?

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

Answer:

A

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

NEW QUESTION 148
- (Exam Topic 4)
Given the below resource configuration - resource "aws_instance" "web" { # ... count = 4 }
What does the terraform resource address aws_instance.web refer to?

A. It refers to all 4 web instances , together , for further individual segregation , indexing is required , with a 0 based index.
B. It refers to the last web EC2 instance , as by default , if no index is provided , the last / N-1 index is used.
C. It refers to the first web EC2 instance out of the 4 ,as by default , if no index is provided , the first / 0th index is used.
D. The above will result in a syntax error , as it is not syntactically correct . Resources defined using count , can only be referenced using indexes.

Answer: A

Explanation:
A Resource Address is a string that references a specific resource in a larger infrastructure. An address is made up of two parts:
[module path][resource spec] Module path:
A module path addresses a module within the tree of modules. It takes the form: module.A.module.B.module.C...
Multiple modules in a path indicate nesting. If a module path is specified without a resource spec, the address applies to every resource within the module. If the module path is omitted, this addresses the root module.
Given a Terraform config that includes: resource "aws_instance" "web" {
# ...
count = 4
}
An address like this: aws_instance.web[3]
Refers to only the last instance in the config, and an address like this: aws_instance.web
Refers to all four "web" instances. https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION 149
- (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 150
- (Exam Topic 4)
When using providers that require the retrieval of data, such as the HashiCorp Vault provider, in what phase does Terraform actually retrieve the data required?

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

Answer: C

NEW QUESTION 152
- (Exam Topic 4)
Which task does teraform init not perform?

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

Answer: B

NEW QUESTION 154
- (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 156
- (Exam Topic 4)
You just upgraded the version of a provider in an existing Terraform project. What do you need to do to install the new provider?

A. Run terraform apply -upgrade
B. Run terraform init -upgrade
C. Run terraform refresh
D. Upgrade your version of Terraform

Answer: B

Explanation:
[-upgrade] - Opt to upgrade modules and plugins as part of their respective installation steps. See the sections below for more details. Reference: https://www.terraform.io/cli/commands/init#upgrade

NEW QUESTION 158
- (Exam Topic 4)
Terraform variable names are saved in the state file.

A. True
B. False

Answer: B

Explanation:
Terraform stores information about your infrastructure in a state file. This state file keeps track of resources created by your configuration and maps them to real-world resources. https://learn.hashicorp.com/tutorials/terraform/state-cli

NEW QUESTION 163
- (Exam Topic 4)
You have created a main.tr Terraform configuration consisting of an application server, a database, and a load balancer. You ran terraform apply and all resources were created successfully. Now you realize that you do not actually need the load balancer so you run terraform destroy without any flags What will happen?

A. Terraform will destroy the application server because it is listed first in the code
B. Terraform will prompt you to confirm that you want to destroy all the infrastructure
C. Terraform will destroy the main.tf file
D. Terraform will prompt you to pick which resource you want to destroy
E. Terraform will immediately destroy all the infrastructure

Answer: B

NEW QUESTION 166
- (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 170
- (Exam Topic 4)
What is the result of the following terraform function call?

A. hello
B. what?
C. goodbye

Answer: B

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

NEW QUESTION 174
- (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 179
- (Exam Topic 4)
Your team uses terraform OSS . You have created a number of resuable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option/way to do that?

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

Answer: B

Explanation:
Software development encourages code reuse through reusable artifacts, such as libraries, packages and modules. Most programming languages enable developers to package and publish these reusable components and make them available on a registry or feed. For example, Python has Python Package Index and PowerShell has PowerShell Gallery.
For Terraform users, the Terraform Registry enables the distribution of Terraform modules, which are reusable configurations. The Terraform Registry acts as a centralized repository for module sharing, making modules easier to discover and reuse.
The Registry is available in two variants:
* Public Registry houses official Terraform providers -- which are services that interact with an API to expose and manage a specific resource -- and community-contributed modules.
* Private Registry is available as part of the Terraform Cloud, and can host modules internally within an organization.
https://www.terraform.io/docs/registry/index.html

NEW QUESTION 184
- (Exam Topic 4)
Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?

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

Answer: A

Explanation:
The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.
https://www.terraform.io/docs/commands/console.html

NEW QUESTION 186
- (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 188
- (Exam Topic 4)
When using multiple configurations of the same Terraform provider, what meta-argument must be included in any non-default provider configurations?

A. name
B. alias
C. depends_on
D. id

Answer: B

NEW QUESTION 192
- (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 196
- (Exam Topic 4)
All Terraform Cloud tiers support team management and governance.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/cloud-docs/overview
Terraform Cloud is a commercial SaaS product developed by HashiCorp. Many of its features are free for small teams, including remote state storage, remote runs, and VCS connections. We also offer paid plans for larger teams that include additional collaboration and governance features. Each higher paid upgrade plan is a strict superset of any lower plans — for example, the Team & Governance plan includes all of the features of the Team plan.

NEW QUESTION 201
- (Exam Topic 4)
In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to use what?

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

Answer: A

Explanation:
Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.
https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION 203
- (Exam Topic 4)
terraform validate validate validates that your infrastructure matches the Terraform state file.

A. True
B. False

Answer: B

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

NEW QUESTION 208
- (Exam Topic 4)
From the answers below, select the advantages of using Infrastructure as Code.

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

Answer: BCDE

Explanation:
Infrastructure as Code is not used to develop applications, but it can be used to help deploy or provision those applications to a public cloud provider or on-premises infrastructure.
All of the others are benefits to using Infrastructure as Code over the traditional way of managing infrastructure, regardless if it's public cloud or on-premises.

NEW QUESTION 209
- (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 211
- (Exam Topic 4)
Which of the following is a meta-argument defined in the configuration files of Terraform?

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

Answer: B

NEW QUESTION 214
- (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 217
- (Exam Topic 4)
Select all Operating Systems that Terraform is available for. (select five)

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

Answer: ABDEF

Explanation:

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

NEW QUESTION 221
- (Exam Topic 4)
What kind of configuration block will create an infrastructure object with settings specified in the block?

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

Answer: C

NEW QUESTION 223
- (Exam Topic 4)
You cannot install third party plugins using terraform init.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/cli/commands/init
For providers that are published in either the public Terraform Registry or in a third-party provider registry, terraform init will automatically find, download, and install the necessary provider plugins.

NEW QUESTION 225
- (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 227
- (Exam Topic 4)
During a terraform plan, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

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

Answer: D

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

NEW QUESTION 229
- (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 233
- (Exam Topic 4)
What is a downside to using the Vault provider to read secrets from Vault?

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

Answer: A

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

NEW QUESTION 236
- (Exam Topic 4)
You need to specify a dependency manually. What resource meta-parameter can you use lo make sure Terraform respects thee dependency?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
depends_on

NEW QUESTION 237
- (Exam Topic 4)
When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)

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

Answer: BCD

Explanation:
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.
With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments: https://www.terraform.io/docs/backends/init.html#backend-initialization

NEW QUESTION 238
- (Exam Topic 4)
Complete the following sentence:
The terraform state command can be used to ______

A. modify state
B. view state
C. refresh state
D. There is no such command

Answer: A

Explanation:

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

NEW QUESTION 239
- (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 240
- (Exam Topic 4)
A junior admin accidentally deleted some of your cloud instances. What does Terraform do when you run terraform apply?

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

Answer: C

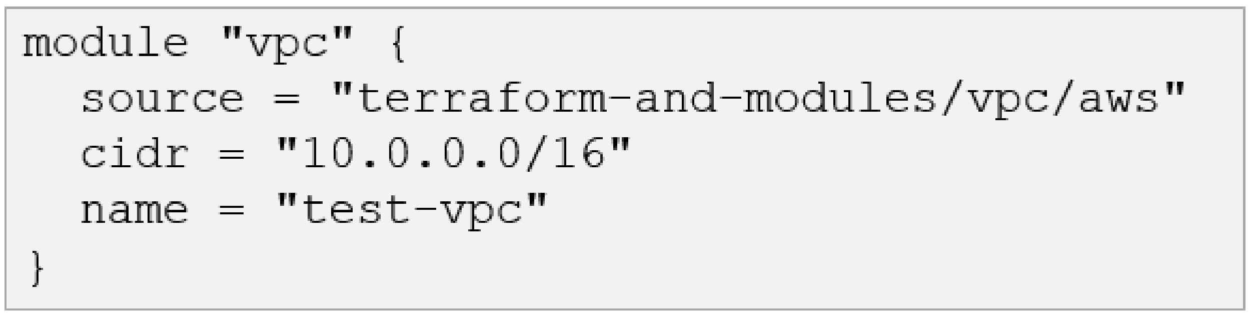
NEW QUESTION 243
- (Exam Topic 4)
Which of the following is not valid source path for specifying a module?

A. source = "./modulelversion=v1.0.0"
B. source = "github.com/hashicorp/example?ref=v1.0.0"
C. source = "./module"

D. source = "hashicorp/consul/aws"

Answer: A

NEW QUESTION 246
- (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 249
- (Exam Topic 4)
True or False? By default, Terraform destroy will prompt for confirmation before proceeding.

A. False
B. True

Answer: B

NEW QUESTION 252
- (Exam Topic 4)
What does terraform refresh modify?

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

Answer: B

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

NEW QUESTION 254
- (Exam Topic 4)
Which of the following connection types are supported by the remote-exec provisioner? (select two)

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

Answer: AE

Explanation:
The remote-exec provisioner invokes a script on a remote resource after it is created. The remote-exec provisioner supports both ssh and winrm type connections.
remote-exec connection types
* ssh on Linux
* winrm on Windows https://www.terraform.io/docs/provisioners/remote-exec.html

NEW QUESTION 255
......

Relate Links

100% Pass Your TA-002-P Exam with Exambible Prep Materials

https://www.exambible.com/TA-002-P-exam/

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - https://www.exambible.com/