

HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate



NEW QUESTION 1

- (Exam Topic 1)

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

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

Answer: C

Explanation:

terraform workspace list

The command will list all existing workspaces.

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

NEW QUESTION 2

- (Exam Topic 1)

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

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

Answer: A

Explanation:

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

NEW QUESTION 3

- (Exam Topic 1)

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)

You have used Terraform to create an ephemeral development environment in the cloud and are now ready to destroy all the infrastructure described by your Terraform configuration. To be safe, you would like to first see all the infrastructure that will be deleted by Terraform.

Which command should you use to show all of the resources that will be deleted? (Choose two.)

- A. Run terraform plan -destroy.
- B. This is not possible
- C. You can only show resources that will be created.
- D. Run terraform state rm *.
- E. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval.

Answer: AD

Explanation:

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

NEW QUESTION 5

- (Exam Topic 1)

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

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

Answer: B

Explanation:

The variable declaration can also include a default argument.

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

NEW QUESTION 6

- (Exam Topic 1)

You should store secret data in the same version control repository as your Terraform configuration.

A. True

B. False

Answer: B

Explanation:

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

NEW QUESTION 7

- (Exam Topic 1)

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

A. True

B. False

Answer: B

Explanation:

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

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

NEW QUESTION 8

- (Exam Topic 1)

FILL BLANK

What is the name of the default file where Terraform stores the state?

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

A. Mastered

B. Not Mastered

Answer: A

Explanation:

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

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

State

JUMP TO SECTION ▾

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

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

NEW QUESTION 9

- (Exam Topic 1)

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

A. True

B. False

Answer: B

Explanation:

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

NEW QUESTION 10

- (Exam Topic 1)

Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)

- A. Destroy
- B. Apply
- C. Import
- D. Init
- E. Validate

Answer: BD

Explanation:

Reference: <https://www.terraform.io/guides/core-workflow.html>

NEW QUESTION 10

- (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 13

- (Exam Topic 1)

Terraform provisioners that require authentication can use the _____ block.

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

Answer: A

Explanation:

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

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

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

NEW QUESTION 14

- (Exam Topic 1)

You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (files). You need to enable debug messages to find this out.

Which of the following would achieve this?

- A. Set the environment variable `TF_LOG=TRACE`
- B. Set verbose logging for each provider in your Terraform configuration
- C. Set the environment variable `TF_VAR_log=TRACE`
- D. Set the environment variable `TF_LOG_PATH`

Answer: A

Explanation:

Although this will only output to stderr and if you need to review log file you will need to include `TF_LOG_PATH=pathToFile`

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

NEW QUESTION 17

- (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 19

- (Exam Topic 1)

One remote backend configuration always maps to a single remote workspace.

- A. True
- B. False

Answer: B

Explanation:

The remote backend can work with either a single remote Terraform Cloud workspace, or with multiple similarly-named remote workspaces (like networking-dev and networking-prod). The workspaces block of the backend configuration determines which mode it uses: To use a single remote Terraform Cloud workspace, set workspaces.name to the remote workspace's full name (like networking-prod). To use multiple remote workspaces, set workspaces.prefix to a prefix used in all of the desired remote workspace names. For example, set prefix = "networking-" to use Terraform cloud workspaces with names like networking-dev and networking-prod. This is helpful when mapping multiple Terraform CLI workspaces used in a single Terraform configuration to multiple Terraform Cloud workspaces.

NEW QUESTION 22

- (Exam Topic 1)

Where does the Terraform local backend store its state?

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

Answer: C

Explanation:

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

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

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

NEW QUESTION 26

- (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 27

- (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 31

- (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 36

- (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 37

- (Exam Topic 1)

When using a module block to reference a module stored on the public Terraform Module Registry such as:

```
module "consul" {  
    source = "hashicorp/consul/aws"  
}
```

How do you specify version 1.0.0?

- A. Modules stored on the public Terraform Module Registry do not support versioning
- B. Append ?ref=v1.0.0 argument to the source path
- C. Add version = "1.0.0" attribute to module block
- D. Nothing – modules stored on the public Terraform Module Registry always default to version 1.0.0

Answer: C

Explanation:

Version

When using modules installed from a module registry, we recommend explicitly constraining the acceptable version numbers to avoid unexpected or unwanted changes.

Use the version argument in the module block to specify versions:

```
module "consul" {  
    source = "hashicorp/consul/aws" version = "0.0.5"  
    servers = 3  
}
```

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

NEW QUESTION 40

- (Exam Topic 1)

A Terraform local value can reference other Terraform local values.

- A. True
- B. False

Answer: A

Explanation:

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

NEW QUESTION 42

- (Exam Topic 1)

Setting the TF_LOG environment variable to DEBUG causes debug messages to be logged into syslog.

- A. True
- B. False

Answer: B

Explanation:

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

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

NEW QUESTION 45

- (Exam Topic 1)

Which of these is the best practice to protect sensitive values in state files?

- A. Blockchain
- B. Secure Sockets Layer (SSL)
- C. Enhanced remote backends
- D. Signed Terraform providers

Answer: C

Explanation:

Use of remote backends and especially the availability of Terraform Cloud, there are now a variety of backends that will encrypt state at rest and will not store the state in cleartext on machines running. Reference:

<https://www.terraform.io/docs/extend/best-practices/sensitive-state.html>

NEW QUESTION 47

- (Exam Topic 1)

When using Terraform to deploy resources into Azure, which scenarios are true regarding state files? (Choose two.)

- A. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in a new state file
- B. When a change is made to the resources via the Azure Cloud Console, Terraform will update the state file to reflect them during the next plan or apply
- C. When a change is made to the resources via the Azure Cloud Console, the current state file will not be updated
- D. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in the current state file

Answer: BC

NEW QUESTION 50

- (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 54

- (Exam Topic 1)

FILL BLANK

You need to specify a dependency manually.

What resource meta-parameter can you use to make sure Terraform respects the dependency?

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 57

- (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 58

- (Exam Topic 1)

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

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

Answer: B

Explanation:

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

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

NEW QUESTION 63

- (Exam Topic 1)

Your DevOps team is currently using the local backend for your Terraform configuration. You would like to move to a remote backend to begin storing the state file in a central location. Which of the following backends would not work?

- A. Amazon S3
- B. Artifactory
- C. Git
- D. Terraform Cloud

Answer: C

Explanation:

<https://www.terraform.io/cdktf/concepts/remote-backends> https://docs.gitlab.com/ee/user/infrastructure/iac/terraform_state.html

NEW QUESTION 68

- (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 71

- (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 75

- (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 76

- (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 78

- (Exam Topic 2)

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

- A. TF_TRACE
- B. TF_DEBUG
- C. TF_LOG
- D. TF_INFO

Answer: C

Explanation:

Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr. You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name. <https://www.terraform.io/docs/internals/debugging.html>

NEW QUESTION 82

- (Exam Topic 2)

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

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

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

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

Answer: B

NEW QUESTION 86

- (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 87

- (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 90

- (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 91

- (Exam Topic 2)

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

- A. False
- B. True

Answer: B

Explanation:

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

NEW QUESTION 92

- (Exam Topic 2)

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

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

Answer: C

Explanation:

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

NEW QUESTION 93

- (Exam Topic 2)

terraform refresh will update the state file?

- A. True
- B. False

Answer: A

Explanation:

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

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

NEW QUESTION 96

- (Exam Topic 2)

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

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

Answer: D

Explanation:

Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure.

Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.

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

NEW QUESTION 101

- (Exam Topic 2)

When using remote state, state is only ever held in memory when used by Terraform.

- A. False
- B. True

Answer: B

NEW QUESTION 106

- (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 109

- (Exam Topic 2)

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

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

Answer: BE

Explanation:

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

NEW QUESTION 112

- (Exam Topic 2)

Which of the following type of variable allows multiple values of several distinct types to be grouped together as a single value?

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

Answer: BC

Explanation:

Structural type of variable allows multiple values of several distinct types to be grouped together as a single value. They require a schema as an argument, to specify which types are allowed for which elements.
<https://www.terraform.io/docs/configuration/types.html>

NEW QUESTION 116

- (Exam Topic 2)

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

- A. export TF_VAR_VAR1
- B. set TF_VAR_var1
- C. variable "var1" { type = "string" }
- D. export TF_VAR_var1

Answer: D

Explanation:

The environment variable must be in the format TF_VAR_name, so for the QUESTION NO: TF_VAR_var1 is the correct choice.
https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION 120

- (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 125

- (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 128

- (Exam Topic 3)

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

"\${var.instance_id}"

- A. variable.instance_id
- B. var.instance_ids
- C. var.instance_id
- D. None of the above

Answer: C

NEW QUESTION 129

- (Exam Topic 3)

You have been given requirements to create a security group for a new application. Since your organization standardizes on Terraform, you want to add this new security group with the fewest number of lines of code. What feature could you use to iterate over a list of required tcp ports to add to the new security group?

- A. dynamic backend
- B. splat expression
- C. terraform import
- D. dynamic block

Answer: D

Explanation:

A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value and generates a nested block for each element of that complex value.

<https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks>

NEW QUESTION 133

- (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 134

- (Exam Topic 3)

Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?

- A. False
- B. True

Answer: A

Explanation:

Terraform module need not be public and open-source. Module can be placed in

- * Local paths
- * Terraform Registry
- * GitHub
- * Bitbucket
- * Generic Git, Mercurial repositories
- * HTTP URLs
- * S3 buckets
- * GCS buckets <https://www.terraform.io/docs/modules/sources.html>

NEW QUESTION 135

- (Exam Topic 3)

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

- A. \${google_storage_bucket.backend}
- B. \${azurerm_resource_group.test.data}
- C. \${aws_instance.web.id.data}
- D. \${data.google_dns_keys.foo_dns_keys.key_signing_keys[0].ds_record}

Answer: D

Explanation:

Data source attributes are interpolated with the general syntax data.TYPE.NAME.ATTRIBUTE. The interpolation for a resource is the same but without the data. prefix (TYPE.NAME.ATTRIBUTE).
<https://www.terraform.io/docs/configuration-0-11/interpolation.html#attributes-of-a-data-source>

NEW QUESTION 140

- (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 143

- (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 148

- (Exam Topic 3)

Your manager has instructed you to start using terraform for your day-to-day operations, but your security team is concerned about the terraform state files. They have heard it contains confidential information, and are worried that it will not be securely protected. What should be your response to the security team in this regard?

- A. Inform the security team that using terraform state is optional . There are ways to avoid it , and you will do the same.
- B. Ensure that the state is managed in a remote backend , preferably an enterprise grade state management system like Terraform Cloud.
- C. Mask the confidential entries in the terraform state file , using Vault Enterprise, another Hashicorp product , while keeping it locally.
- D. Keep the state file locally on each developer machine , and ensure that there is a local protection software like KeyPass protecting it.

Answer: B

Explanation:

<https://www.terraform.io/docs/state/index.html>
State is very important topic for exam. Please read all of the below subtopics
Purpose
Import Existing Resources
Locking
Workspaces
Remote State
Sensitive Data

NEW QUESTION 150

- (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 155

- (Exam Topic 3)

State locking does not happen automatically and must be specified at run

A. False

B. True

Answer: A

Explanation:

State locking happens automatically on all operations that could write state. <https://www.terraform.io/docs/state/locking.html>

NEW QUESTION 156

- (Exam Topic 3)

You cannot publish your own modules on the Terraform Registry.

A. False

B. True

Answer: A

Explanation:

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

NEW QUESTION 159

- (Exam Topic 3)

Which flag would be used within a Terraform configuration block to identify the specific version of a provider required?

A. required-provider

B. required-version

C. required_providers

D. required_versions

Answer: C

Explanation:

For production use, you should constrain the acceptable provider versions via configuration file to ensure that new versions with breaking changes will not be automatically installed by terraform init in the future.

```
Example terraform {  
  required_providers { aws = ">= 2.7.0"  
}  
}
```

NEW QUESTION 162

- (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 163

- (Exam Topic 3)

Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance.

Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

A. terraform recreate

B. terraform taint

C. terraform destroy

D. terraform refresh

Answer: B

Explanation:

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

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

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

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

This example will taint a single resource:

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

The resource aws_security_group.allow_all in the module root has been marked as tainted. <https://www.terraform.io/docs/commands/taint.html>

NEW QUESTION 166

- (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 170

- (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 173

- (Exam Topic 3)

You also have a defined the following environment variables in your shell: TF_itemNum =6, TF_VAR_itemNum =9. You also have a terraform.tfvars file with the following contents

```
itemNum = 7
```

When you run the following apply command, what is the value assigned to the itemNum variable? terraform apply -var itemNum =4

- A. 10
- B. 6
- C. 1
- D. 4
- E. 3

Answer: D

Explanation:

The -var and -var-file methods of assigning variables have the highest precedence. <https://www.terraform.io/docs/configuration/variables.html>

NEW QUESTION 176

- (Exam Topic 3)

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

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

Answer: B

Explanation:

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

NEW QUESTION 178

- (Exam Topic 3)

Terraform-specific settings and behaviors are declared in which configuration block type?

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

Answer: B

Explanation:

The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.

```
Example terraform {  
  required_version = "> 0.12.0"  
}
```

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

NEW QUESTION 179

- (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 182

- (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 184

- (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 185

- (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 187

- (Exam Topic 3)

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

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

Answer: A

Explanation:

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

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

NEW QUESTION 189

- (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 194

- (Exam Topic 4)

terraform init retrieves the source code for all referenced modules

- A. True
- B. False

Answer: A

Explanation:

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

NEW QUESTION 197

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 199

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

NEW QUESTION 202

- (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 204

- (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 208

- (Exam Topic 4)

Select the most accurate statement to describe the Terraform language from the following list.

- A. Terraform is an immutable, declarative, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally JSON.
- B. Terraform is a mutable, declarative, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
- C. Terraform is an immutable, procedural, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
- D. Terraform is a mutable, procedural, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally YAML.

Answer: A

Explanation:

Terraform is not a configuration management tool - <https://www.terraform.io/intro/vs/chefpuppet.html> Terraform is a declarative language - <https://www.terraform.io/docs/configuration/index.html> Terraform supports a syntax that is JSON compatible <https://www.terraform.io/docs/configuration/syntax-json.html>
Terraform is primarily designed on immutable infrastructure principles - <https://www.hashicorp.com/resources/what-is-mutable-vs-immutable-infrastructure>

NEW QUESTION 210

- (Exam Topic 4)

What does the command terraform fmt do?

- A. Rewrite Terraform configuration files to a canonical format and style.
- B. Deletes the existing configuration file.
- C. Updates the font of the configuration file to the official font supported by HashiCorp.
- D. Formats the state file in order to ensure the latest state of resources can be obtained.

Answer: A

Explanation:

The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style. This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability.
Other Terraform commands that generate Terraform configuration will produce configuration files that conform to the style imposed by terraform fmt, so using this style in your own files will ensure consistency.
<https://www.terraform.io/docs/commands/fmt.html>

NEW QUESTION 215

- (Exam Topic 4)

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

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

Answer: D

NEW QUESTION 216

- (Exam Topic 4)

You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name the instance resource "main".

Which of these is the correct way to define the output value using HCL2?

A.

```
output "instance_ip_addr" {  
    value = "${aws_instance.main.private_ip}"  
}
```

B.

```
output "instance_ip_addr" {  
    return aws_instance.main.private_ip  
}
```

A. Option A

B. Option B

Answer: A

NEW QUESTION 221

- (Exam Topic 4)

In the example below, the depends_on argument creates what type of dependency?

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

Answer: C

NEW QUESTION 222

- (Exam Topic 4)

To check if all code in a Terraform configuration with multiple modules is properly formatted without making changes, what command should be run?

- A. terraform fmt -check
- B. terraform fmt -write=false
- C. terraform fmt "list -recursive
- D. terraform fmt -check -recursive

Answer: D

Explanation:

-check Check if the input is formatted. Exit status will be 0 if all input is properly formatted and non-zero otherwise.

-recursive Also process files in subdirectories. By default, only the given directory (or current directory) is processed.

NEW QUESTION 226

- (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 228

- (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 230

- (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 235

- (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 239

- (Exam Topic 4)

Why should secrets not be hard coded into Terraform code? Choose two correct answers

- A. All passwords should be rotated on a quarterly basis.
- B. The Terraform code is copied to the target resources to be applied locally and could expose secrets if a target resource is compromised.
- C. Terraform code is typically stored in version control, as well as copied to the systems from h it's run.Any of those may not have robust security mechanisms.
- D. It makes the code less reusable.

Answer: BC

NEW QUESTION 241

- (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 246

- (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 250

- (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 253

- (Exam Topic 4)

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

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

Answer: CE

NEW QUESTION 256

- (Exam Topic 4)

From the code below, identify the implicit dependency:

- A. The EIP with an id of ami-2757f631
- B. The AMI used for the EC2 instance
- C. The EC2 instance labeled web_server
- D. The S3 bucket labeled company_data

Answer: C

NEW QUESTION 260

- (Exam Topic 4)

Multiple provider instances blocks for AWS can be part of a single configuration file?

- A. False
- B. True

Answer: B

Explanation:

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

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

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

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

NEW QUESTION 261

- (Exam Topic 4)

State is a requirement for Terraform to function

- A. True
- B. False

Answer: A

Explanation:

State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run.

Purpose of Terraform State

State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.

As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).

* 1. Mapping to the Real World

Terraform requires some sort of database to map Terraform config to the real world. When you have a resource resource "aws_instance" "foo" in your configuration, Terraform uses this map to know that instance i- abcd1234 is represented by that resource.

For some providers like AWS, Terraform could theoretically use something like AWS tags. Early prototypes of Terraform actually had no state files and used this method. However, we quickly ran into problems. The first major issue was a simple one: not all resources support tags, and not all cloud providers support tags.

Therefore, for mapping configuration to resources in the real world, Terraform uses its own state structure.

* 2. Metadata

Alongside the mappings between resources and remote objects, Terraform must also track metadata such as resource dependencies.

Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.

To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.

One way to avoid this would be for Terraform to know a required ordering between resource types. For example, Terraform could know that servers must be deleted before the subnets they are a part of. The

complexity for this approach quickly explodes, however: in addition to Terraform having to understand the ordering semantics of every resource for every cloud, Terraform must also understand the ordering across providers.

Terraform also stores other metadata for similar reasons, such as a pointer to the provider configuration that was most recently used with the resource in situations where multiple aliased providers are present.

* 3. Performance

In addition to basic mapping, Terraform stores a cache of the attribute values for all resources in the state. This is the most optional feature of Terraform state and is done only as a performance improvement.

When running a terraform plan, Terraform must know the current state of resources in order to effectively determine the changes that it needs to make to reach your desired configuration.

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.

* 4. Syncing

In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but when using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects.

Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION 263

- (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 264

- (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 269

- (Exam Topic 4)

Which Terraform command will check and report errors within modules, attribute names, and value types to make sure they are syntactically valid and internally consistent?

- A. terraform validate
- B. terraform format
- C. terraform fmt
- D. terraform show

Answer: A

Explanation:

The terraform validate command validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs, etc.

Validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus primarily useful for general verification of reusable modules, including the correctness of attribute names and value types.

It is safe to run this command automatically, for example as a post-save check in a text editor or as a test step for a re-usable module in a CI system.

NEW QUESTION 272

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 274

- (Exam Topic 4)

True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

- A. False
- B. True

Answer: B

Explanation:

The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

NEW QUESTION 275

- (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 276

- (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 279

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 284

- (Exam Topic 4)

By default, where does Terraform store its state file?

- A. Amazon S3 bucket
- B. shared directory
- C. remotely using Terraform Cloud
- D. current working directory

Answer: D

Explanation:

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

NEW QUESTION 287

- (Exam Topic 4)

A single terraform resource file that defines an `aws_instance` resource can simple be renamed to `azurerm_virtual_machine` in order to switch cloud providers

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 289

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

NEW QUESTION 294

- (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 295

- (Exam Topic 4)

You can reference a resource created with `for_each` using a Splat (*) expression.

- A. True
- B. False

Answer: B

Explanation:

Splat Expressions with Maps The splat expression patterns shown above apply only to lists, sets, and tuples. To get a similar result with a map or object value you must use `for_each` for expressions. Resources that use the `for_each` argument will appear in expressions as a map of objects, so you can't use splat expressions with those resources. For more information, see Referring to Resource Instances. https://www.terraform.io/language/meta-arguments/for_each#referring-to-instances

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

NEW QUESTION 299

- (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 300

- (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 302

- (Exam Topic 4)

terraform apply will fail if you have not run terraform plan first to update the plan output.

- A. True
- B. False

Answer: B

NEW QUESTION 306

- (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 307

- (Exam Topic 4)

Which of the following is not supported backend types in Terra form?

- A. consul
- B. gcs
- C. manta
- D. bitbucket

Answer: D

NEW QUESTION 308

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 310

- (Exam Topic 4)

In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?

- A. Resource dependencies are identified and maintained in a file called resource.dependencie
- B. Each terraform provider is required to maintain a list of all resource dependencies for the provider and it's included with the plugin during initialization when terraform init is execute
- C. The file is located in the terraform.d folder.
- D. The terraform binary contains a built-in reference map of all defined Terraform resource dependencies.Updates to this dependency map are reflected in terraform version
- E. To ensure you are working with thelatest resource dependency map you much be running the latest version of Terraform.
- F. Resource dependencies are handled automatically by the depends_on meta_argument, which is set to true by default.
- G. Terraform analyses any expressions within a resource block to find references to other objects, and treats those references as implicit ordering requirements when creating, updating, or destroying resources.

Answer: D

Explanation:

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

NEW QUESTION 311

- (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 312

- (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 316

- (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 317

- (Exam Topic 4)

While Terraform is generally written using the HashiCorp Configuration Language (HCL), what other syntax can Terraform are expressed in?

- A. JSON
- B. YAML
- C. TypeScript
- D. XML

Answer: A

Explanation:

The constructs in the Terraform language can also be expressed in JSON syntax, which is harder for humans to read and edit but easier to generate and parse programmatically.

NEW QUESTION 319

- (Exam Topic 4)

If a DevOps team adopts AWS Cloud Formation as their standardized method for provisioning public cloud resources, which of the following scenarios poses a challenge for this team?

- A. The team is asked to manage a new application stack built on AWS-natrve services
- B. The organization decides to expand into Azure and wishes to deploy new infrastructure using their existing codebase
- C. The team is asked to build a reusable code base that can deploy resources into any AWS region
- D. The DevOps team is tasked with automating a manual provisioning process

Answer: B

NEW QUESTION 321

- (Exam Topic 4)

Using multi-cloud and provider-agnostic tools provides which of the following benefits?

- A. Operations teams only need to learn and manage a single tool to manage infrastructure, regardless of where the infrastructure is deployed.
- B. Increased risk due to all infrastructure relying on a single tool for management.
- C. Can be used across major cloud providers and VM hypervisors.
- D. Slower provisioning speed allows the operations team to catch mistakes before they are applied.

Answer: AC

Explanation:

Using a tool like Terraform can be advantageous for organizations deploying workloads across multiple public and private cloud environments. Operations teams only need to learn a single tool, single language, and can use the same tooling to enable a DevOps-like experience and workflows.

NEW QUESTION 324

- (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 326

- (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 328

- (Exam Topic 4)

What does terraform refresh modify?

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

Answer: C

NEW QUESTION 331

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 334

- (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 338

- (Exam Topic 4)

What are some of the problems of how infrastructure was traditionally managed before Infrastructure as Code? (select three)

- A. Requests for infrastructure or hardware required a ticket, increasing the time required to deploy applications
- B. Traditional deployment methods are not able to meet the demands of the modern business where resources tend to live days to weeks, rather than months to years
- C. Traditionally managed infrastructure can't keep up with cyclic or elastic applications
- D. Pointing and clicking in a management console is a scalable approach and reduces human error as businesses are moving to a multi-cloud deployment model

Answer: ABC

Explanation:

Businesses are making a transition where traditionally-managed infrastructure can no longer meet the demands of today's businesses. IT organizations are quickly adopting the public cloud, which is predominantly API-driven. To meet customer demands and save costs, application teams are architecting their applications to support a much higher level of elasticity, supporting technology like containers and public cloud resources. These resources may only live for a matter of hours; therefore the traditional method of raising a ticket to request resources is no longer a viable option. Pointing and clicking in a management console is NOT scale and increases the change of human error.

NEW QUESTION 340

- (Exam Topic 4)

You need to specify a dependency manually. What resource meta-parameter can you use to make sure Terraform respects the dependency?

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:
depends_on

NEW QUESTION 344

- (Exam Topic 4)

Once a new Terraform backend is configured with a Terraform code block, which command(s) is (are) used to migrate the state file?

- A. terraform apply
- B. terraform push
- C. terraform destroy, then terraform apply
- D. terraform init

Answer: B

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

NEW QUESTION 348

- (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 351

- (Exam Topic 4)

You need to write some Terraform code that adds 42 firewall rules to a security group as shown in the example.

```
resource "aws_security_group" "many_rules" {
  name = "many-rules"
  ingress {
    from_port = 443
    to_port = 443
    protocol = "tcp"
    cidr_blocks = "0.0.0.0/0"
  }
}
```

What can you use to avoid writing 42 different nested ingress config blocks by hand?

- A. A count loop
- B. A for block
- C. A for each block
- D. A dynamic block

Answer: D

Explanation:
A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value, and generates a nested block for each element of that complex value. Reference: <https://www.terraform.io/language/expressions/dynamic-blocks>

NEW QUESTION 352

- (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 357

- (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 358

- (Exam Topic 4)

terraform validate reports HCL syntax errors.

- A. True
- B. False

Answer: A

NEW QUESTION 362

- (Exam Topic 4)

What advantage does an operations team that uses infrastructure as code have?

- A. The ability to delete infrastructure
- B. The ability to reuse best practice configurations and settings
- C. The ability to autoscale a group of servers
- D. The ability to update existing infrastructure

Answer: B

NEW QUESTION 367

- (Exam Topic 4)

A module can always refer to all variables declared in its parent module.

- A. True
- B. False

Answer: B

Explanation:

Modules do not inherit variables from the parent module. All modules are self-contained units. So you have to explicitly define variables in the child module, and then explicit set these variables in the parent module, when you instantiate the child module.

NEW QUESTION 368

- (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 371

- (Exam Topic 4)

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

- A. Vault provider
- B. API access using the AppRole auth method

- C. integration with a tool like Jenkins
- D. CLI access from the same machine running Terraform

Answer: A

NEW QUESTION 373

- (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 374

- (Exam Topic 4)

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

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

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

Answer: B

Explanation:

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

NEW QUESTION 376

- (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 381

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

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

NEW QUESTION 385

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 389

- (Exam Topic 4)

You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration?

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 394

- (Exam Topic 4)

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

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

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

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

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

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

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

Answer: C

Explanation:

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

NEW QUESTION 395

- (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 400

- (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 402

- (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 407

- (Exam Topic 4)

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

- A. replace
- B. format
- C. join
- D. tostring

Answer: D

Explanation:

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

NEW QUESTION 412

- (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 413

- (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 418

.....

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](#)