

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)

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

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

Answer: C

Explanation:

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

NEW QUESTION 3

- (Exam Topic 1)

Terraform provisioners can be added to any resource block.

- A. True
- B. False

Answer: A

Explanation:

<https://www.phillipsj.net/posts/introduction-to-terraform-provisioners/>

As you continue learning about Terraform, you will start hearing about provisioners. Terraform provisioners can be created on any resource and provide a way to execute actions on local or remote machines.

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

NEW QUESTION 4

- (Exam Topic 1)

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 5

- (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 6

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 7

- (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 8

- (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 9

- (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 10

- (Exam Topic 1)

Which of these options is the most secure place to store secrets for connecting to a Terraform remote backend?

- A. Defined in Environment variables
- B. Inside the backend block within the Terraform configuration
- C. Defined in a connection configuration outside of Terraform
- D. None of above

Answer: A

Explanation:

<https://www.terraform.io/language/settings/backends/configuration#credentials-and-sensitive-data> Warning: We recommend using environment variables to supply

credentials and other sensitive data. If you use -backend-config or hardcode these values directly in your configuration, Terraform will include these values in both the .terraform subdirectory and in plan files. This can leak sensitive credentials.

NEW QUESTION 10

- (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 13

- (Exam Topic 1)

Only the user that generated a plan may apply it.

- A. True
- B. False

Answer: B

NEW QUESTION 15

- (Exam Topic 1)

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

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

Answer: A

Explanation:

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

NEW QUESTION 16

- (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 17

- (Exam Topic 1)

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

- A. True
- B. False

Answer: A

Explanation:

Output values are like function return values.

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

NEW QUESTION 18

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

NEW QUESTION 23

- (Exam Topic 1)

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

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

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

Answer: D

Explanation:

version = ">= 1.2.0, < 2.0.0"

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

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

The following operators are valid:

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

!=: Excludes an exact version number.

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

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

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

NEW QUESTION 26

- (Exam Topic 1)

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

- A. Add node_count = var.node_count
- B. Declare the variable in a terraform.tfvars file
- C. Declare a node_count input variable for child module
- D. Nothing, child modules inherit variables of parent module

Answer: C

Explanation:

"That module may call other modules and connect them together by passing output values from one to input values of another."

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

NEW QUESTION 28

- (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 29

- (Exam Topic 1)

What is not processed when running a terraform refresh?

- A. State file
- B. Configuration file
- C. Credentials
- D. Cloud provider

Answer: B

Explanation:

"The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match."

NEW QUESTION 31

- (Exam Topic 1)

A Terraform provisioner must be nested inside a resource configuration block.

- A. True
- B. False

Answer: A

Explanation:

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

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

NEW QUESTION 32

- (Exam Topic 1)

What does the default "local" Terraform backend store?

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

Answer: D

Explanation:

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

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

NEW QUESTION 35

- (Exam Topic 1)

Which of the following is not a valid Terraform collection type?

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

Answer: C

Explanation:

<https://www.terraform.io/language/expressions/type-constraints#collection-types>

NEW QUESTION 37

- (Exam Topic 1)

A Terraform provider is not responsible for:

- A. Understanding API interactions with some service
- B. Provisioning infrastructure in multiple clouds
- C. Exposing resources and data sources based on an API
- D. Managing actions to take based on resource differences

Answer: B

Explanation:

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

NEW QUESTION 39

- (Exam Topic 1)

What is terraform refresh intended to detect?

- A. Terraform configuration code changes
- B. Empty state files
- C. State file drift
- D. Corrupt state files

Answer: C

Explanation:

"The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match. Warning: This command is deprecated, because its default behavior is unsafe if you have misconfigured credentials for any of your providers. See below for more information and recommended alternatives." <https://www.terraform.io/cli/commands/refresh>

NEW QUESTION 43

- (Exam Topic 1)

All standard backend types support state storage, locking, and remote operations like plan. apply and destroy.

- A. True
- B. False

Answer: B

Explanation:

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

"Some of these backends act like plain remote disks for state files, while others support locking the state while operations are being performed. This helps prevent conflicts and inconsistencies. The built-in backends listed are the only backends. You cannot load additional backends as plugins."

NEW QUESTION 48

- (Exam Topic 1)

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

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

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

Answer: A

Explanation:

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

NEW QUESTION 52

- (Exam Topic 2)

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

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

Answer: D

Explanation:

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

NEW QUESTION 54

- (Exam Topic 2)

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

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

Answer: ACD

Explanation:

Terraform is cloud-agnostic and allows a single configuration to be used to manage multiple providers, and to even handle cross-cloud dependencies. This simplifies management and orchestration, helping operators build large-scale multi-cloud infrastructures.

<https://www.terraform.io/intro/use-cases.html>

NEW QUESTION 56

- (Exam Topic 2)

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

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

Answer: C

Explanation:

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

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

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

NEW QUESTION 57

- (Exam Topic 2)

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

- A. False
- B. True

Answer: A

Explanation:

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

NEW QUESTION 62

- (Exam Topic 2)

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

- A. Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using

Policy-As-Code.

B. Implement a review process where every code will be reviewed before merging to the master branch.

C. Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and writePolicy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.

D. Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.

Answer: C

Explanation:

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

NEW QUESTION 64

- (Exam Topic 2)

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

A. provisioner "local-exec" { command = "echo 0" command = "echo 1" }

B. provisioner "remote-exec" { inline = [echo 0,echo 1]}

C. provisioner "remote-exec" {command = "\${echo 0}" command = "\${echo 1}" }

D. provisioner "remote-exec" { inline = ["echo 0","echo 1"] }

Answer: D

Explanation:

remote-exec Provisioner Example usage

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

```
# ...
```

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

```
"puppet apply",
```

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

```
]
```

```
}
```

```
}
```

NEW QUESTION 69

- (Exam Topic 2)

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

A. map

B. set

C. string

D. list

Answer: A

Explanation:

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

NEW QUESTION 73

- (Exam Topic 2)

What is the default backend for Terraform?

A. consul

B. gcs

C. local

D. etcd

Answer: C

Explanation:

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

NEW QUESTION 74

- (Exam Topic 2)

The current implementation of Terraform import can only import resources into the state. It does not generate configuration.

A. False

B. True

Answer: B

Explanation:

The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration.

Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.

While this may seem tedious, it still gives Terraform users an avenue for importing existing resources. <https://www.terraform.io/docs/import/index.html#currently-state-only>

NEW QUESTION 77

- (Exam Topic 2)

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

- A. True
- B. False

Answer: B

Explanation:

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

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

```
variable "variable_name" {}
```

```
terraform apply -var variable_name="value"
```

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

String

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

```
variable "template" { type = string
```

```
default = "01000000-0000-4000-8000-000030080200"
```

```
}
```

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

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

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

Here is an example of a list variable definition.

```
variable "users" { type = list
```

```
default = ["root", "user1", "user2"]
```

```
}
```

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

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

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

```
variable "plans" { type = map default = {
```

```
"5USD" = "1xCPU-1GB" "10USD" = "1xCPU-2GB" "20USD" = "2xCPU-4GB"
```

```
}
```

```
}
```

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

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

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

```
variable "storage_sizes" { type = map
```

```
default = {
```

```
"1xCPU-1GB" = "25"
```

```
"1xCPU-2GB" = "50"
```

```
"2xCPU-4GB" = "80"
```

```
}
```

```
}
```

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

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

Boolean

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

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

```
}
```

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

```
create_password = var.set_password
```

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

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

NEW QUESTION 82

- (Exam Topic 2)

You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?

- A. It depends on which command issue related to.
- B. Python
- C. Go
- D. Java

Answer: C

Explanation:

Basic programming knowledge. Terraform and Terraform Plugins are written in the Go programming language, but even if you've never written a line of Go before, you're still welcome to take a dive into the code and submit patches. The community is happy to assist with code reviews and offer guidance specific to Go.

NEW QUESTION 87

- (Exam Topic 2)

You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?

- A. terraform unlock
- B. terraform force-unlock
- C. terraform state unlock
- D. None of the above

Answer: B

Explanation:

Command: force-unlock

Manually unlock the state for the defined configuration.

This will not modify your infrastructure. This command removes the lock on the state for the current configuration. The behavior of this lock is dependent on the backend being used. Local state files cannot be unlocked by another process.

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

Terraform has a force-unlock command to manually unlock the state if unlocking failed.

If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed.

NEW QUESTION 91

- (Exam Topic 2)

Please identify the offerings which are unique to Terraform Enterprise, and not available in either Terraform OSS, or Terraform Cloud. Select four.

- A. Audit Logs
- B. Private Network Connectivity
- C. VCS Integration
- D. Sentinel
- E. Clustering

Answer: ABE

Explanation:

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

NEW QUESTION 92

- (Exam Topic 2)

What is the purpose of using the local-exec provisioner? (Select Two)

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

Answer: AC

Explanation:

The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource. Note that even though the resource will be fully created when the provisioner is run, there is no guarantee that it will be in an operable state - for example system services such as sshd may not be started yet on compute resources.

Example usage

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

Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives.

<https://www.terraform.io/docs/provisioners/local-exec.html>

NEW QUESTION 96

- (Exam Topic 2)

Terraform must track metadata such as resource dependencies. Where is this data stored?

- A. workspace
- B. backend
- C. state file
- D. metadata store

Answer: C

Explanation:

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

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

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

NEW QUESTION 100

- (Exam Topic 3)

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

- A. terraform state list
- B. terraform state view
- C. terraform view
- D. terraform list

Answer: A

Explanation:

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

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

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

NEW QUESTION 104

- (Exam Topic 3)

Which of the following challenges would Terraform be a candidate for solving? (Select THREE)

- A. Enable self-service infrastructure to allocate resources on your proprietary private cloud.
- B. Reduce the number of workflows needed for managing infrastructure across each of the companies public and private clouds.
- C. Utilize a single tool for all of the infrastructure and configuration management needs.
- D. Have a single interoperable tool to manage the variety of services including GitHub repositories, MySQL database, and Kubernetes clusters.

Answer: ABD

NEW QUESTION 105

- (Exam Topic 3)

You have created two workspaces PROD and DEV. You have switched to DEV and provisioned DEV infrastructure from this workspace. Where is your state file stored?

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

Answer: D

Explanation:

Terraform stores the workspace states in a directory called terraform.tfstate.d. This directory should be treated similarly to default workspace state file

terraform.tfstate main.tf

provider.tf terraform.tfstate.d DEV

terraform.tfstate # DEV workspace state file PROD

terraform.tfstate # PROD workspace state file terraform.tfvars # Default workspace state file variables.tf

NEW QUESTION 109

- (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 114

- (Exam Topic 3)

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

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

Answer: D

NEW QUESTION 117

- (Exam Topic 3)

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

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

D. Upload the modules in the terraform public module registry , and ask teams to reference them

Answer: B

Explanation:

Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster.

By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar.

Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise. Terraform Cloud's private module registry helps you share Terraform modules across your organization.

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

NEW QUESTION 122

- (Exam Topic 3)

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

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

Answer: B

Explanation:

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

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

Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure. When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.

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

NEW QUESTION 124

- (Exam Topic 3)

Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.

- A. Set TF_LOG = DEBUG
- B. Set VAR_TF = TRACE
- C. Set TF_LOG = TRACE
- D. Set VAR_TF_LOG = TRACE

Answer: C

Explanation:

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

NEW QUESTION 125

- (Exam Topic 3)

Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.

- A. terraform state
- B. terraform apply
- C. terraform plan
- D. terraform validate
- E. terraform output

Answer: BC

NEW QUESTION 130

- (Exam Topic 3)

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

- A. False
- B. True

Answer: B

Explanation:

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

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

NEW QUESTION 132

- (Exam Topic 3)

Refer below code where pessimistic constraint operator has been used to specify a version of a provider. terraform { required_providers { aws = "~> 1.1.0" }}

Which of the following options are valid provider versions that satisfy the above constraint. (select two)

- A. 1.1.1
- B. 1.2.9
- C. 1.1.8
- D. 1.2.0

Answer: AC

Explanation:

Pessimistic constraint operator, constraining both the oldest and newest version allowed. For example, ~> 0.9 is equivalent to >= 0.9, < 1.0, and ~> 0.8.4, is equivalent to >= 0.8.4, < 0.9

NEW QUESTION 135

- (Exam Topic 3)

What happens when a terraform apply command is executed?

- A. Creates the execution plan for the deployment of resources.
- B. Applies the changes required in the target infrastructure in order to reach the desired configuration.
- C. The backend is initialized and the working directory is prepped.
- D. Reconciles the state Terraform knows about with the real-world infrastructure.

Answer: B

Explanation:

The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.

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

NEW QUESTION 140

- (Exam Topic 3)

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

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

Answer: B

Explanation:

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

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

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

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

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

NEW QUESTION 141

- (Exam Topic 3)

A data block requests that Terraform read from a given data source and export the result under the given local name.

- A. False
- B. True

Answer: B

NEW QUESTION 146

- (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 148

- (Exam Topic 3)

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

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

Answer: ABCDEF

Explanation:

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

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

Debian 7.7+

Ubuntu 14.04.5 / 16.04 / 18.04

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

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

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

NEW QUESTION 151

- (Exam Topic 3)

You cannot publish your own modules on the Terraform Registry.

- A. False
- B. True

Answer: A

Explanation:

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

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

NEW QUESTION 152

- (Exam Topic 3)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 154

- (Exam Topic 3)

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

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

Answer: A

Explanation:

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

NEW QUESTION 155

- (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 157

- (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 159

- (Exam Topic 4)

Which of the following is an invalid variable name?

- A. count
- B. web
- C. var1
- D. instance_name

Answer: A

Explanation:

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

NEW QUESTION 162

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 167

- (Exam Topic 4)

HashiCorp offers multiple versions of Terraform, including Terraform open-source, Terraform Cloud, and Terraform Enterprise. Which of the following Terraform features are only available in the Enterprise edition? (select four)

- A. SAML/SSO
- B. Sentinel
- C. Audit Logs
- D. Clustering
- E. Private Module Registry
- F. Private Network Connectivity

Answer: ACF

Explanation:

While there are a ton of features that are available to open source users, many features that are part of the Enterprise offering are geared towards larger teams and enterprise functionality. To see what specific features are part of Terraform Cloud and Terraform Enterprise, check out this link.

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

NEW QUESTION 168

- (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 171

- (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 174

- (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 176

- (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 177

- (Exam Topic 4)

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

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

A. resource.kubnrnetes_namespace>example.name

B. kubernetes_namespace.test.name

- C. kubernetes_namespace.example,name
- D. data kubernetes_namespace.name
- E. None of the above

Answer: C

Explanation:

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

NEW QUESTION 182

- (Exam Topic 4)

A user runs terraform init on their RHEL based server and per the output, two provider plugins are downloaded: \$ terraform init

Initializing the backend... Initializing provider plugins...

- Checking for available provider plugins...

- Downloading plugin for provider "aws" (hashicorp/aws) 2.44.0...

- Downloading plugin for provider "random" (hashicorp/random) 2.2.1...

:

Terraform has been successfully initialized! Where are these plugins downloaded to?

- A. The .terraform.plugins directory in the directory terraform init was executed in.
- B. The .terraform/plugins directory in the directory terraform init was executed in.
- C. /etc/terraform/plugins
- D. The .terraform.d directory in the directory terraform init was executed in.

Answer: B

NEW QUESTION 185

- (Exam Topic 4)

Your risk management organization requires that new AWS S3 buckets must be private and encrypted at rest. How can Terraform Enterprise automatically and proactively enforce this security control?

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

Answer: A

Explanation:

<https://docs.hashicorp.com/sentinel/intro/what>

<https://medium.com/hashicorp-engineering/enforcing-aws-s3-security-best-practice-using-terraform-sentinel-dd>

NEW QUESTION 187

- (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 192

- (Exam Topic 4)

Any user can publish modules to the public Terraform Module Registry.

- A. True
- B. False

Answer: B

NEW QUESTION 195

- (Exam Topic 4)

Which of the following arguments are required when declaring a Terraform output?

- A. sensitive
- B. description
- C. default
- D. value

Answer: D

NEW QUESTION 197

- (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 200

- (Exam Topic 4)

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

- A. False
- B. True

Answer: B

NEW QUESTION 203

- (Exam Topic 4)

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

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

Answer: CD

Explanation:

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

NEW QUESTION 207

- (Exam Topic 4)

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

- A. True
- B. False

Answer: A

NEW QUESTION 210

- (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 215

- (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 217

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

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

NEW QUESTION 219

- (Exam Topic 4)

A Terraform backend determines how Terraform loads state and stores updates when you execute ____.

- A. apply
- B. taint
- C. destroy
- D. All of the above
- E. None of the above

Answer: D

NEW QUESTION 224

- (Exam Topic 4)

terraform destroy is the only way to remove infrastructure.

- A. True
- B. False

Answer: B

NEW QUESTION 226

- (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 231

- (Exam Topic 4)

True or False: Workspaces provide identical functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 232

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

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

The Registry is available in two variants:

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

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

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

NEW QUESTION 233

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

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

NEW QUESTION 234

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 236

- (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 241

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 245

- (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 246

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 249

- (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 252

- (Exam Topic 4)

From the answers below, select the advantages of using Infrastructure as Code.

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

Answer: BCDE

Explanation:

Infrastructure as Code is not used to develop applications, but it can be used to help deploy or provision those applications to a public cloud provider or on-premises infrastructure.

All of the others are benefits to using Infrastructure as Code over the traditional way of managing infrastructure, regardless if it's public cloud or on-premises.

NEW QUESTION 254

- (Exam Topic 4)

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

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

Answer: B

NEW QUESTION 255

- (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 260

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 265

- (Exam Topic 4)

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

- A. True
- B. False

Answer: A

Explanation:

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

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

NEW QUESTION 267

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 270

- (Exam Topic 4)

Which of the following is considered a Terraform plugin?

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

Answer: D

Explanation:

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

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

NEW QUESTION 271

- (Exam Topic 4)

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

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

Answer: ABDEF

Explanation:

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

NEW QUESTION 275

- (Exam Topic 4)

You've used Terraform to deploy a virtual machine and a database. You want to replace this virtual machine instance with an identical one without affecting the database. What is the best way to achieve this using Terraform?

- A. Use the Terraform taint command targeting the VMs then run Terraform plan and Terraform apply
- B. Delete the Terraform VM resources from your Terraform code then run Terraform plan and terraform apply
- C. Use the terraform apply command targeting the VM resources only
- D. Use the terraform state rm command to remove the VM from state file

Answer: A

Explanation:

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

NEW QUESTION 276

- (Exam Topic 4)

Terraform plan updates your state file.

- A. True
- B. False

Answer: B

Explanation:

The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review. Source: <https://www.terraform.io/cli/commands/plan>

NEW QUESTION 277

- (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 280

- (Exam Topic 4)

Which are examples of infrastructure as code? (Choose two.)

- A. Cloned virtual machine images
- B. Change management database records
- C. Versioned configuration files
- D. Docker files

Answer: CD

NEW QUESTION 281

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 284

- (Exam Topic 4)

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

- A. Using the consul Terraform backend.
- B. Using the remote Terraform backend with Terraform Cloud / Terraform Enterprise.
- C. Using the local backend.
- D. Using the s3 terraform backen
- E. The dynamodb_field option e not needed.

F. Using an s3 terraform backend with an appropriate IAM policy and dynamodb_field option configured.

Answer: ABE

NEW QUESTION 286

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

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

NEW QUESTION 290

- (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 295

- (Exam Topic 4)

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

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

Answer: A

NEW QUESTION 296

- (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 298

- (Exam Topic 4)

You want to define multiple data disks as nested blocks inside the resource block for a virtual machine. What Terraform feature would help you define the blocks using the values in a variable?

- A. Local values
- B. Dynamic blocks
- C. Count arguments
- D. Collection functions

Answer: B

NEW QUESTION 302

- (Exam Topic 4)

Which of the following does terraform apply change after you approve the execution plan? Choose two correct answers.

- A. The execution plan
- B. Terraform code
- C. Cloud infrastructure
- D. State file
- E. The .terraform directory

Answer: CD

NEW QUESTION 306

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

NEW QUESTION 311

- (Exam Topic 4)

Which of the following is not an advantage of using infrastructure as code operations?

- A. Self-service infrastructure deployment
- B. Troubleshoot via a Linux diff command
- C. Public cloud console configuration workflows
- D. Modify a count parameter to scale resources
- E. API driven workflows

Answer: B

Explanation:

terraform is used to deploy the infrastructure, not to troubleshoot it

NEW QUESTION 314

- (Exam Topic 4)

What does Terraform use providers for? (Choose three.)

- A. Provision resources for on-premises infrastructure services
- B. Simplify API interactions
- C. Provision resources for public cloud infrastructure services
- D. Enforce security and compliance policies
- E. Group a collection of Terraform configuration files that map to a single state file

Answer: ABC

NEW QUESTION 318

- (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 writePolicy-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 321

- (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 325

- (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 329

- (Exam Topic 4)

Which of the following commands will launch the Interactive console for Terraform interpolations?

- A. terraform console
- B. terraform cli
- C. terraform
- D. terraform cmdline

Answer: B

Explanation:

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

NEW QUESTION 334

- (Exam Topic 4)

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

- A. False
- B. True

Answer: B

NEW QUESTION 338

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 341

- (Exam Topic 4)

What Terraform feature is shown in the example below?

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

Answer: C

NEW QUESTION 344

- (Exam Topic 4)

What does terraform destroy do?

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

Answer: A

Explanation:

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

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

NEW QUESTION 347

- (Exam Topic 4)

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

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

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

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

- A. Terraform will not make any changes to the Load Balancer and will update the state file to reflect any changes made.
- B. Terraform will change the port back to 80 in your code
- C. Terraform will change the load balancer port to 80, and) then change it back to 443
- D. Terraform will fail with in error because the state file is no longer accurate

Answer: A

NEW QUESTION 350

.....

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 FirstTry
- * TA-002-P Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year

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