

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)

You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you delete the VM using the cloud provider console, and run terraform apply again without changing any Terraform code?

- A. Terraform will remove the VM from state file
- B. Terraform will report an error
- C. Terraform will not make any changes
- D. Terraform will recreate the VM

Answer: D

NEW QUESTION 4

- (Exam Topic 1)

You write a new Terraform configuration and immediately run terraform apply in the CLI using the local backend. Why will the apply fail?

- A. Terraform needs you to format your code according to best practices first
- B. Terraform needs to install the necessary plugins first
- C. The Terraform CLI needs you to log into Terraform cloud first
- D. Terraform requires you to manually run terraform plan first

Answer: B

NEW QUESTION 5

- (Exam Topic 1)

When should you use the force-unlock command?

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

Answer: C

Explanation:

Be very careful with this command. If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed. Source: <https://www.terraform.io/language/state/locking>
<https://www.terraform.io/cli/commands/force-unlock>

NEW QUESTION 6

- (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 7

- (Exam Topic 1)

terraform validate validates the syntax of Terraform files.

- A. True
- B. False

Answer: A

Explanation:

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

The terraform validate command validates the syntax and arguments of the Terraform configuration files. Reference:

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

NEW QUESTION 8

- (Exam Topic 1)

You need to deploy resources into two different cloud regions in the same Terraform configuration. To do that, you declare multiple provider configurations as follows:

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

What meta-argument do you need to configure in a resource block to deploy the resource to the "us-west-2" AWS region?

- A. alias = west
- B. provider = west
- C. provider = aws.west
- D. alias = aws.west

Answer: C

Explanation:

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

NEW QUESTION 9

- (Exam Topic 1)

Terraform can only manage resource dependencies if you set them explicitly with the depends_on argument.

- A. True
- B. False

Answer: A

Explanation:

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

https://www.terraform.io/language/meta-arguments/depends_on

NEW QUESTION 10

- (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 10

- (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 15

- (Exam Topic 1)

Which task does terraform init not perform?

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

Answer: D

Explanation:

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

NEW QUESTION 19

- (Exam Topic 1)

Examine the following Terraform configuration, which uses the data source for an AWS AMI. What value should you enter for the ami argument in the AWS instance resource?

```
data "aws_ami" "ubuntu" {
  ...
}

resource "aws_instance" "web" {
  ami = _____
  instance_type = "t2.micro"

  tags = {
    Name = "HelloWorld"
  }
}
```

- A. aws_ami.ubuntu
- B. data.aws_ami.ubuntu
- C. data.aws_ami.ubuntu.id
- D. aws_ami.ubuntu.id

Answer: C

Explanation:

resource "aws_instance" "web" { ami= data.aws_ami.ubuntu.id
Reference: <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance>

NEW QUESTION 23

- (Exam Topic 1)

What value does the Terraform Cloud/Terraform Enterprise private module registry provide over the public Terraform Module Registry?

- A. The ability to share modules with public Terraform users and members of Terraform Enterprise Organizations
- B. The ability to tag modules by version or release
- C. The ability to restrict modules to members of Terraform Cloud or Enterprise organizations
- D. The ability to share modules publicly with any user of Terraform

Answer: C

Explanation:

Terraform Cloud's private registry works similarly to the public Terraform Registry and helps you share Terraform providers and Terraform modules across your organization. It includes support for versioning and a searchable list of available providers and modules.

NEW QUESTION 25

- (Exam Topic 1)

A provider configuration block is required in every Terraform configuration. Example:

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 26

- (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 30

- (Exam Topic 1)

Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/downloads>

NEW QUESTION 33

- (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 35

- (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 40

- (Exam Topic 1)

What features does the hosted service Terraform Cloud provide? (Choose two.)

- A. Automated infrastructure deployment visualization
- B. Automatic backups
- C. Remote state storage
- D. A web-based user interface (UI)

Answer: CD

Explanation:

<https://www.terraform.io/enterprise/admin/infrastructure/backup-restore>

NEW QUESTION 43

- (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 48

- (Exam Topic 1)

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

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

Answer: D

Explanation:

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

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

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

NEW QUESTION 50

- (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 52

- (Exam Topic 1)

How would you reference the "name" value of the second instance of this fictitious resource?

```
resource "aws_instance" "web" {  
  count = 2  
  name = "terraform-${count.index}"  
}
```

- A. element(aws_instance.web, 2)
- B. aws_instance.web[1].name
- C. aws_instance.web[1]
- D. aws_instance.web[2].name
- E. aws_instance.web.*.name

Answer: B

Explanation:

<https://www.terraform.io/language/meta-arguments/count#referring-to-instances> Reference: <https://www.terraform.io/docs/configuration-0-11/interpolation.html>

NEW QUESTION 55

- (Exam Topic 1)

FILL BLANK

Which flag would you add to terraform plan to save the execution plan to a file?

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:

"You can use the optional -out=FILE option to save the generated plan to a file on disk, which you can later execute by passing the file to terraform apply as an extra argument. This two-step workflow is primarily intended for when running Terraform in automation. If you run terraform plan without the -out=FILE option then it will create a speculative plan, which is a description of the effect of the plan but without any intent to actually apply it." <https://www.terraform.io/cli/commands/plan>

NEW QUESTION 58

- (Exam Topic 1)

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

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

Answer: B

Explanation:

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

NEW QUESTION 61

- (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 64

- (Exam Topic 1)

When does terraform apply reflect changes in the cloud environment?

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

Answer: B

NEW QUESTION 68

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 70

- (Exam Topic 1)

HashiCorp Configuration Language (HCL) supports user-defined functions.

- A. True
- B. False

Answer: B

Explanation:

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

The Terraform language does not support user-defined functions, and so only the functions built into the language are available for use

NEW QUESTION 71

- (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 73

- (Exam Topic 1)

Terraform requires the Go runtime as a prerequisite for installation.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/plugin/sdkv2/guides/v1-upgrade-guide> and <https://www.terraform.io/plugin/sdkv2/guides/v2-upgrade-guide>

NEW QUESTION 74

- (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 79

- (Exam Topic 1)

If writing Terraform code that adheres to the Terraform style conventions, how would you properly indent each nesting level compared to the one above it?

- A. With four spaces
- B. With a tab
- C. With three spaces
- D. With two spaces

Answer: D

Explanation:

<https://www.terraform.io/language/syntax/style#style-conventions>

NEW QUESTION 80

- (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 83

- (Exam Topic 1)

Terraform and Terraform providers must use the same major version number in a single configuration.

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 88

- (Exam Topic 1)

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

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

- A. compute_instance
- B. main
- C. google
- D. test

Answer: B

NEW QUESTION 89

- (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 93

- (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 98

- (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 101

- (Exam Topic 1)

Which of the following is not true of Terraform providers?

- A. Providers can be written by individuals
- B. Providers can be maintained by a community of users
- C. Some providers are maintained by HashiCorp
- D. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers
- E. None of the above

Answer: E

Explanation:

<https://registry.terraform.io/providers/hashicorp/google/latest> - This provider is collaboratively maintained by the Google Terraform Team at Google and the Terraform team at HashiCorp

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

NEW QUESTION 106

- (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 107

- (Exam Topic 1)

You run a local-exec provisioner in a null resource called null_resource.run_script and realize that you need to rerun the script.

Which of the following commands would you use first?

- A. terraform taint null_resource.run_script
- B. terraform apply -target=null_resource.run_script
- C. terraform validate null_resource.run_script
- D. terraform plan -target=null_resource.run_script

Answer: A

Explanation:

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

NEW QUESTION 110

- (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 114

- (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 117

- (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 120

- (Exam Topic 1)

Which backend does the Terraform CLI use by default?

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

Answer: D

Explanation:

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

NEW QUESTION 125

- (Exam Topic 2)

Which one of the following command will rewrite Terraform configuration files to a canonical format and style.

- A. terraform graph -h
- B. terraform init
- C. terraform graph
- D. terraform fmt

Answer: D

Explanation:

The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style. This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability.

NEW QUESTION 129

- (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 runs 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 130

- (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 134

- (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 138

- (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 139

- (Exam Topic 2)

ABC Enterprise has recently tied up with multiple small organizations for exchanging database information. Due to this, the firewall rules are increasing and are more than 100 rules. This is leading to a firewall configuration file that is difficult to manage. What is the way this type of configuration can be managed easily?

- A. Terraform Backends
- B. Terraform Functions
- C. Dynamic Blocks
- D. Terraform Expression

Answer: C

NEW QUESTION 144

- (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 146

- (Exam Topic 2)

You have created 2 workspaces PROD and RQA. You have switched to RQA and provisioned RQA infrastructure from this workspace. Where is your state file stored?

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

Answer: A

NEW QUESTION 149

- (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 150

- (Exam Topic 2)

terraform refresh command will not modify infrastructure, but does modify 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.

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

NEW QUESTION 152

- (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 155

- (Exam Topic 2)

By default, a defined provisioner is a creation-time provisioner.

- A. True
- B. False

Answer: A

Explanation:

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

NEW QUESTION 157

- (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 160

- (Exam Topic 2)

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

- A. False
- B. True

Answer: A

Explanation:

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

NEW QUESTION 165

- (Exam Topic 2)

Which of the following command can be used to view the specified version constraints for all providers used in the current configuration.

- A. terraform providers
- B. terraform state show
- C. terraform provider
- D. terraform plan

Answer: A

Explanation:

Use the terraform providers command to view the specified version constraints for all providers used in the current configuration.
<https://www.terraform.io/docs/configuration/providers.html>

NEW QUESTION 169

- (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 171

- (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 175

- (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 180

- (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 181

- (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 185

- (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 189

- (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 191

- (Exam Topic 3)

Dawn has created the below child module. Without changing the module, can she override the instance_type from t2.micro to t2.large from her code while calling this module?

```
* 1. resource "aws_instance" "myec2"  
* 2. {  
* 3. ami = "ami-082b5a644766e0e6f"  
* 4. instance_type = "t2.micro"  
* 5. }
```

- A. YES
- B. No

Answer: B

Explanation:

As the instance_type is hard-coded in source module, you will not be able to change its value from destination module. Instead of hard-coding you should use variable with default values.

NEW QUESTION 193

- (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 194

- (Exam Topic 3)

The Security Operations team of ABC Enterprise wants to mandate that all the Terraform configuration that creates an S3 bucket must have encryption feature

enabled. What is the best way to achieve it?

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

Answer: A

Explanation:

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

Using Sentinel with Terraform Cloud involves:

- * Defining the policies - Policies are defined using the policy language with imports for parsing the Terraform plan, state and configuration.
- * Managing policies for organizations - Users with permission to manage policies can add policies to their organization by configuring VCS integration or uploading policy sets through the API. They also define which workspaces the policy sets are checked against during runs. (More about permissions.)
- * Enforcing policy checks on runs - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is executed.
- * Mocking Sentinel Terraform data - Terraform Cloud provides the ability to generate mock data for any run within a workspace. This data can be used with the Sentinel CLI to test policies before deployment.

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

NEW QUESTION 196

- (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 201

- (Exam Topic 3)

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

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

Answer: C

Explanation:

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

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

NEW QUESTION 203

- (Exam Topic 3)

Which of the below commands will rename a EC2 instance without destroying and recreating it?

- A. terraform state mv
- B. terraform mv
- C. terraform plan
- D. terraform plan mv

Answer: A

NEW QUESTION 208

- (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 213

- (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 216

- (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 218

- (Exam Topic 3)

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

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

Answer: A

Explanation:

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

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

NEW QUESTION 219

- (Exam Topic 3)

The terraform state command can be used to _____

- A. Update current state
- B. Refresh existing state file
- C. Print the current state file in console
- D. It is not a valid command

Answer: A

Explanation:

The terraform state command is used for advanced state management. Rather than modify the state directly, the terraform state commands can be used in many cases instead.

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

NEW QUESTION 221

- (Exam Topic 3)

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

- A. False
- B. True

Answer: A

Explanation:

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

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

NEW QUESTION 226

- (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 228

- (Exam Topic 3)

Mary has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code:

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

Mary wants to hide the output value in the CLI after terraform apply? What is the best way?

- A. Use secure parameter
- B. Use sensitive parameter
- C. Use cryptographic hash
- D. Encrypt the value using encrypt() function

Answer: B

NEW QUESTION 229

- (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 230

- (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 235

- (Exam Topic 3)

A user has created three workspaces using the command line - prod, dev, and test. The user wants to create a fourth workspace named stage. Which command will the user execute to accomplish this?

- A. terraform workspace new stage
- B. terraform workspace -new stage
- C. terraform workspace -create stage
- D. terraform workspace create stage

Answer: A

Explanation:

The terraform workspace new command is used to create a new workspace. <https://www.terraform.io/docs/commands/workspace/new.html>

NEW QUESTION 239

- (Exam Topic 3)

You have provisioned some aws resources in your test environment through Terraform for a POC work. After the POC, now you want to destroy the resources but before destroying them you want to check what resources will be getting destroyed through terraform. what are the options of doing that? (Select TWO)

- A. Use terraform destroy command
- B. This is not possible

- C. Use terraform plan command
- D. Use terraform plan -destroy command.

Answer: AD

Explanation:

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

NEW QUESTION 241

- (Exam Topic 3)

Refer to the following terraform variable definition

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

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

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

Answer: B

NEW QUESTION 242

- (Exam Topic 4)

How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? (Choose two.)

- A. A full audit trail of the request and fulfillment process is generated
- B. A request must be submitted for infrastructure changes
- C. As additional resources are required, more tickets are submitted
- D. A catalog of approved resources can be accessed from drop down lists in a request form

Answer: BC

NEW QUESTION 247

- (Exam Topic 4)

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

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

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

- A. Yes
- B. No

Answer: A

Explanation:

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

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

Limitations of Sensitive Outputs

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

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

NEW QUESTION 252

- (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 253

- (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 257

- (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 260

- (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 264

- (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 265

- (Exam Topic 4)

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

- A. Initializes downloaded and/or installed providers
- B. Initializes the backend configuration
- C. Provisions the declared resources in your configuration
- D. Download the declared providers which are supported by HashiCorp

Answer: ABD

Explanation:

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

- * Child Module Installation
- * Plugin Installation <https://www.terraform.io/docs/commands/init.html>

NEW QUESTION 270

- (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 272

- (Exam Topic 4)

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

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

Answer: BC

Explanation:

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

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

NEW QUESTION 274

- (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 279

- (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 284

- (Exam Topic 4)

Terra form installs its providers during which phase?

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

Answer: B

Explanation:

Providers are installed in the init phase

NEW QUESTION 286

- (Exam Topic 4)

Which of the following can you do with terraform plan? Choose two correct answers.

- A. View the execution plan and check if the changes match your expectations
- B. Schedule Terraform to run at a planned time in the future
- C. Execute a plan in a different workspace
- D. Save a generated execution plan to apply later

Answer: AD

Explanation:

<https://learn.hashicorp.com/tutorials/terraform/plan>

NEW QUESTION 287

- (Exam Topic 4)

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

Original configuration:

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

Updated configuration:

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

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

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

Answer: C

Explanation:

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

NEW QUESTION 289

- (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 292

- (Exam Topic 4)

When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)

- * 1. terraform
- * 2. {
- * 3. required_providers
- * 4. {
- * 5. aws = "~> 1.2.0"
- * 6. }
- * 7. }

A. 1.3.1

- B. 1.2.3
- C. 1.2.9
- D. 1.3.0

Answer: BC

Explanation:

As your Terraform usage becomes more advanced, there are some cases where you may need to modify the Terraform state. Rather than modify the state directly, the terraform state commands can be used in many cases instead. This command is a nested subcommand, meaning that it has further subcommands.
<https://www.terraform.io/docs/commands/state/index.html>

NEW QUESTION 296

- (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 298

- (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 301

- (Exam Topic 4)

You just upgraded the version of a provider in an existing Terraform project. What do you need to do to install the new provider?

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

Answer: B

Explanation:

[-upgrade] - Opt to upgrade modules and plugins as part of their respective installation steps. See the sections below for more details. Reference:

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

NEW QUESTION 306

- (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 307

- (Exam Topic 4)

You're preparing to install Terraform on client workstations and want to see which operating systems are supported. Which of the following operating systems is supported?

- A. Windows
- B. Amazon Linux
- C. FreeBSD
- D. Solaris
- E. MacOS
- F. All of the above

Answer: F

NEW QUESTION 309

- (Exam Topic 4)

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

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

Answer: B

NEW QUESTION 310

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

NEW QUESTION 314

- (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 315

- (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 318

- (Exam Topic 4)

colleagues is new to Terraform and wants to add a new workspace named new-hire. What command he should execute from the following?

- A. terraform workspace-new-new-hire
- B. terraform workspace new new hire
- C. terraform workspace init new-hire
- D. terraform workspace new-hire

Answer: B

NEW QUESTION 319

- (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 323

- (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 325

- (Exam Topic 4)

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

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

Answer: B

NEW QUESTION 329

- (Exam Topic 4)

A single terraform resource file that defines an aws_instance resource can simple be renamed to azurevm_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 330

- (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 332

- (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 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 335

- (Exam Topic 4)

How does Terraform determine dependencies between resources?

- A. Terraform automatically builds a resource graph based on resources, provisioners, special meta-parameters, and the state file, if present.
- B. Terraform requires all dependencies between resources to be specified using the `depends_on` parameter
- C. Terraform requires resources in a configuration to be listed in the order they will be created to determine dependencies
- D. Terraform requires resource dependencies to be defined as modules and sourced in order

Answer: A

Explanation:

<https://learn.hashicorp.com/tutorials/terraform/dependencies>

NEW QUESTION 339

- (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 342

- (Exam Topic 4)

Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.

What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

- A. Submit a ticket to AWS and ask them to export the state of all existing resources and use terraform import to import them into the state file.
- B. Delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward.
- C. Resources that are manually deployed in the AWS console cannot be imported by Terraform.
- D. Using terraform import, import the existing infrastructure into your Terraform state.

Answer: D

Explanation:

Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.

This is a great way to slowly transition infrastructure to Terraform.

The terraform import command is used to import existing infrastructure.

To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.

Example:

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

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

```
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
```

```
aws_instance.import_example: Import complete!
```

```
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
```

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside

Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 345

- (Exam Topic 4)

All Terraform Cloud tiers support team management and governance.

- A. True
- B. False

Answer: B

Explanation:

<https://www.terraform.io/cloud-docs/overview>

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

NEW QUESTION 347

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 352

- (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 353

- (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 354

- (Exam Topic 4)

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

- A. tfvar
- B. depends_on
- C. instance aws
- D. var!

Answer: B

NEW QUESTION 355

- (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 358

- (Exam Topic 4)

Select the feature below that best completes the sentence:

The following list represents the different types of _____ available in Terraform.

- * 1. max
- * 2. min
- * 3. join
- * 4. replace
- * 5. list
- * 6. length
- * 7. range

- A. Backends
- B. Data sources
- C. Named values
- D. Functions

Answer: D

Explanation:

The Terraform language includes a number of built-in functions that you can call from within expressions to transform and combine values. The Terraform language does not support user-defined functions, and only the functions built into the language are available for use.

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

NEW QUESTION 359

- (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 364

- (Exam Topic 4)

What kind of configuration block will create an infrastructure object with settings specified in the block?

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

Answer: C

NEW QUESTION 368

- (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 373

- (Exam Topic 4)

When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.

For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.

- A. Export the value using terraform export and input the value using terraform input.
- B. Configure the pertinent provider's configuration with a list of possible IP addresses to use.
- C. Configure an output value in the application module in order to use that value for the DNS module.
- D. Preconfigure the IP address as a parameter in the DNS module.

Answer: C

Explanation:

Output values are like the return values of a Terraform module, and have several uses:

- * A child module can use outputs to expose a subset of its resource attributes to a parent module.
 - * A root module can use outputs to print certain values in the CLI output after running terraform apply.
 - * When using remote state, root module outputs can be accessed by other configurations via a terraform_remote_state data source.
- <https://www.terraform.io/docs/configuration/outputs.html>

NEW QUESTION 378

- (Exam Topic 4)

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

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

Answer: D

NEW QUESTION 380

- (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 382

- (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 383

- (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 385

- (Exam Topic 4)

Which of the following is true about terraform apply? (Choose two.)

- A. It only operates on infrastructure defined in the current working directory or workspace
- B. You must pass the output of a terraform plan command to it
- C. Depending on provider specification, Terraform may need to destroy and recreate your infrastructure resources
- D. By default, it does not refresh your state file to reflect current infrastructure configuration
- E. You cannot target specific resources for the operation

Answer: AC

Explanation:

<https://www.terraform.io/cli/run>

NEW QUESTION 387

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

NEW QUESTION 392

- (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 394

- (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 399

- (Exam Topic 4)

A variable az has the following default value. What will be the datatype of the variable? az=["us-west-1a","us-east-1a"]

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

Answer: B

NEW QUESTION 403

- (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 407

- (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 411

- (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 416

- (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 418

- (Exam Topic 4)

What command can you run to generate DOT (Document Template) formatted data to visualize Terraform dependencies?

- A. terraform refresh
- B. terraform show
- C. terraform graph
- D. terraform output

Answer: C

Explanation:

The terraform graph command is used to generate a visual representation of either a configuration or execution plan. The output is in the DOT format, which can be used by GraphViz to generate charts.

NEW QUESTION 420

- (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 422

- (Exam Topic 4)

Choose the best option from below to make Terraform code more user configuration-centric.

- A. Variables
- B. Local values
- C. Input Variable
- D. Modules

Answer: C

NEW QUESTION 424

- (Exam Topic 4)

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

- A. False
- B. True

Answer: B

NEW QUESTION 425

- (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 428

- (Exam Topic 4)

terraform apply is failing with the following error. What next step should you take to determine the root cause of the problem?
Error loading state: AccessDenied: Access Denied status code: 403, request id: 288766CE5CCA24A0, host id: FOOBAR

- A. Set TF_LOG=DEBUG
- B. Review syslog for Terraform error messages
- C. Run terraform login to reauthenticate with the provider
- D. Review /var/log/terraform.log for error messages

Answer: A

Explanation:

Terraform has detailed logs which can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr. You can set TF_LOG to one of the log levels (in order of decreasing verbosity) TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.

NEW QUESTION 431

- (Exam Topic 4)

In the following code snippet, the block type is identified by which string?

- A. "aws_instance"
- B. resource
- C. "db"
- D. instance_type

Answer: B

NEW QUESTION 432

- (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 436

- (Exam Topic 4)

Which of the below backends support state locking?

- A. S3
- B. consul
- C. azurearm
- D. artifactory

Answer: ABC

NEW QUESTION 437

- (Exam Topic 4)

As a member of an operations team that uses infrastructure as code (IaC) practices, you are tasked with making a change to an infrastructure stack running in a public cloud. Which pattern would follow IaC best practices for making a change?

- A. Make the change via the public cloud API endpoint
- B. Make the change programmatically via the public cloud CLI
- C. Submit a pull request and wait for an approved merge of the proposed changes
- D. Use the public cloud console to make the change after a database record has been approved
- E. Clone the repository containing your infrastructure code and then run the code

Answer: C

NEW QUESTION 438

- (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 441

- (Exam Topic 4)

Most Terraform providers interact with _____.

- A. API
- B. VCS Systems
- C. Shell scripts
- D. None of the above

Answer: A

Explanation:

Terraform relies on plugins called "providers" to interact with cloud providers, SaaS providers, and other APIs, as per: <https://www.terraform.io/language/providers>

NEW QUESTION 442

.....

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](#)