Question #1 *Topic 1*

IaC (Infrastructure as Code) can be stored in a version control system along with application code.

A. True

B. False

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #2 *Topic 1*

It is best practice to store secret data in the same version control repository as your Terraform configuration.

A. True

B. False

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #3 *Topic 1*

Which is an advantage of using IaC (Infrastructure as Code) that is not possible when provisioning with a GUI (Graphical User Interface)?

A. Let's you version, reuse, and share infrastructure configuration.

B. Secures your credentials.

C. Provisions the same resources at a lower cost.

D. Prevents manual modifications to your resources.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #4 *Topic 1*

What is an advantage of immutable infrastructure?

A. Automatic infrastructure upgrades

B. In-place infrastructure upgrades

C. Quicker infrastructure upgrades

D. Less complex infrastructure upgrades

Suggested Answer: *D*

Community vote distribution

C (100%)

ykamal 1 day, 17 hours ago

Selected Answer: D

AI said its D.

upvoted 1 times

Srikantha 3 days, 8 hours ago

Selected Answer: D

D. Less complex infrastructure upgrades is the correct advantage of immutable infrastructure.

Immutable infrastructure treats servers, VMs, or resources as immutable (unchanging once deployed). To apply updates, fixes, or changes, you create entirely new instances from a known, versioned image or template and then replace (or decommission) the old ones, rather than modifying them in place. This contrasts with mutable infrastructure, where you update existing resources directly (e.g., via patches, config changes, or SSH).

upvoted 1 times

mai_will 3 weeks, 1 day ago

Selected Answer: C

Embora "mais rápido" possa parecer subjetivo, refere-se à previsibilidade e à velocidade de implementação inerentes ao modelo imutável. Em vez de acessar servidores e executar scripts de atualização que podem travar ou falhar, você simplesmente substitui a versão antiga por uma nova pré-configurada.

upvoted 1 times

Question #5 *Topic 1*

What is the primary purpose of IaC (Infrastructure as Code)?

A. To define a pipeline to test and deliver software.

B. To provision infrastructure cheaply.

C. To programmatically create and configure resources.

D. To define a vendor-agnostic API.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #6 *Topic 1*

Your team adopts an AWS CloudFormation as the standardized method for provisioning public cloud resources.
Which scenario presents a challenge for your team?

A. Deploying new infrastructure into Microsoft Azure.

B. Automating a manual, web console-based provisioning process.

C. Building a reusable code base that can deploy resources into any AWS region.

D. Managing a new application stack built on AWS-native services.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #7 *Topic 1*

Which is not a benefit of adopting IaC (Infrastructure as Code)?

A. Reusability of code

B. Automation

C. A GUI (Graphical User Interface)

D. Versioning

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #8 *Topic 1*

How does the use of Infrastructure as Code (IaC) enhance the reliability of your infrastructure? (Choose two.)

A. Configuration drift is reduced with declarative definitions.

B. Updates are deployed with zero downtime.

C. Proposed changes can be reviewed before being applied.

D. Incorrect configurations cannot be deployed.

E. Infrastructure is automatically scaled to meet demand.

Suggested Answer: *AC*

Currently there are no comments in this discussion, be the first to comment!

Question #9 *Topic 1*

Your team often uses API calls to create and manage cloud infrastructure.

In what ways does Terraform differ from conventional infrastructure management approaches?

A. Terraform replaces cloud provider APIs with its own protocols, enabling automated deployments.

B. Terraform describes infrastructure with version-controlled, repeatable configurations that specify the desired state.

C. Terraform is merely a wrapper for cloud provider APIs, so there is little to no difference in calling the API directly.

D. Terraform enforces infrastructure through imperative scripts to ensure tasks are completed in the proper order.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #10 *Topic 1*

Which of these workflows is only enabled by the use of Infrastructure as Code?

A. Automatic scaling of resources based on application load.

B. Cost optimization of infrastructure deployment.

C. Role-based access control of cloud resources.

D. Reviewing the proposed changes for potential security issues.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #11 *Topic 1*

What does Terraform use to deploy infrastructure for different cloud providers?

A. Custom APIs developed by HashiCorp

B. Vendors' CLI tools

C. Vendors' UI

D. Vendor-specific providers

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #1 *Topic 2*

How can you enable verbose logging to troubleshoot?

A. Set the log level command line flag.

B. Set the TF_LOG environment variable.

C. Set the log level in your terraform block.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #2 *Topic 2*

Which command lets you experiment with Terraform expressions?

A. terraform console

B. terraform env

C. terraform validate

D. terraform test

Suggested Answer: *A*

Srikantha 3 days, 8 hours ago

Selected Answer: A

This is the correct command for experimenting with Terraform expressions.

upvoted 1 times

Question #3 *Topic 2*

You can install Community/Partner plugins using terraform init.

A. True

B. False

Suggested Answer: *A*

Srikantha 3 days, 8 hours ago

Selected Answer: A

terraform init does install Community and Partner plugins (providers).

upvoted 1 times

Question #4 *Topic 2*

What functionality do providers offer in Terraform? (Choose three.)

A. Group a collection of Terraform configuration files that map to a single state file.

B. Provision resources for public cloud infrastructure services.

C. Interact with cloud provider APIs.

D. Enforce security and compliance policies.

E. Provision resources for on-premises infrastructure services.

Suggested Answer: *BCE*

Srikantha 3 days, 8 hours ago

Selected Answer: BCE

B. Provision resources for public cloud infrastructure services.
Providers (e.g., aws, azurerm, google) let you create, update, and delete resources in AWS, Azure, GCP, etc.
C. Interact with cloud provider APIs.
This is the core job of a provider: it handles authentication, makes the actual API calls (create/read/update/delete), and translates Terraform's declarative configuration into real infrastructure changes.
E. Provision resources for on-premises infrastructure services.
Providers also support self-hosted/on-prem platforms (e.g., vsphere for VMware, openstack, bare-metal providers, Kubernetes, databases, etc.).
Terraform is not limited to public cloud.

upvoted 1 times

Question #5 *Topic 2*

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

A. True

B. False

Suggested Answer: *B*

Srikantha 3 days, 8 hours ago

Selected Answer: B

Terraform can (and usually does) manage resource dependencies automatically through implicit dependencies — no depends_on required in most cases.

upvoted 1 times

Question #6 *Topic 2*

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

A. set

B. list

C. map

D. tree

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #7 *Topic 2*

A resource block is shown in the Exhibit space of this page.
resource “aws_vpc” “main” {
name = “test”
}
What is the provider for this resource?

A. main

B. vpc

C. aws

D. test

Suggested Answer: *C*

Srikantha 2 days, 17 hours ago

Selected Answer: C

C. aws
The provider for this resource is aws.
How Terraform identifies the provider
In a Terraform resource block, the syntax is always:
hclresource "<RESOURCE_TYPE>" "<RESOURCE_NAME>" { ... }

RESOURCE_TYPE = aws_vpc
The part before the first underscore (aws) tells Terraform which provider is responsible for this resource.
The part after the underscore (vpc) is the specific resource type offered by that provider.
RESOURCE_NAME = main
This is just a local identifier (used for referencing the resource inside your configuration). It has nothing to do with the provider.
upvoted 1 times

Question #8 *Topic 2*

You must use different Terraform commands depending on the cloud provider you use.

A. True

B. False

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #9 *Topic 2*

Terraform providers are always installed from the internet.

A. True

B. False

Suggested Answer: *B*

Srikantha 2 days, 17 hours ago

Selected Answer: B

B. False

Terraform providers are not always installed from the internet. While the default behavior (when you run terraform init) is to download providers from the public Terraform Registry at registry.terraform.io (or a private registry), Terraform provides built-in support for offline / air-gapped installations through provider mirrors.

upvoted 1 times

Question #10 *Topic 2*

You need to determine from which paths Terraform is loading the providers referenced in your *.tf files.
How can you enable additional logging to see this information?

A. Set the environment variable TF_VAR_log=TRACE.

B. Set the environment variable TF_LOG=TRACE.

C. Set the environment variable TF_LOG=PATH.

D. Set verbose logging for each provider in your Terraform configuration.

Suggested Answer: *B*

Srikantha 2 days, 17 hours ago

Selected Answer: B

B. Set the environment variable TF_LOG=TRACE.
This is the official, recommended way to enable detailed logging in Terraform so you can see exactly which filesystem paths it is searching for and loading providers from (including the plugin cache, filesystem mirrors, dev overrides, and any custom directories)

upvoted 1 times

Question #11 *Topic 2*

Terraform requires using a different provider for each cloud provider where you want to deploy resources.

A. True

B. False

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #12 *Topic 2*

Terraform cannot use a newly-defined cloud backend until it has been initialized with terraform init.

A. True

B. False

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #13 *Topic 2*

You have a list of numbers that represents the number of free CPU cores on each virtual cluster: numcpus = [18, 3, 7, 11, 2]

What Terraform built-in function would you use to select the largest number from the list?

A. top(numcpus)

B. max(numcpus)

C. ceil(numcpus)

D. high[numcpus]

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #14 *Topic 2*

Which of these is stored in the .terraform directory?

A. Providers and modules

B. Lock file

C. State file

D. Configuration files

Suggested Answer: *A*

Community vote distribution

A (100%)

Srikantha 2 days, 16 hours ago

Selected Answer: A

A. Providers and modules

The hidden .terraform directory (created automatically by terraform init) is Terraform's local cache for providers and modules.

upvoted 1 times

Question #15 *Topic 2*

A variable block is shown in the Exhibit space on this page:

```
variable "tags" {
  description = "Metadata tags for resources"
  type = _____________
}
```

You will use this variable as the value for the tags argument in several resources. The data format must be a set of key value pairs.

Which type argument would you use?

A. type = map(string)

B. type = any

C. type = list(string)

D. type = object({ tags = string})

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #16 *Topic 2*

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

A. name

B. source

C. version

D. count

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #1 *Topic 3*

terraform init creates an example main.tf file in the current directory.

A. True

B. False

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #2 *Topic 3*

Which command must you run before you run a plan or apply for the first time?

A. terraform validate

B. terraform workspace

C. terraform import

D. terraform init

Suggested Answer: *D*

Srikantha 2 days, 16 hours ago

Selected Answer: D

terraform init is the mandatory first command you must run in any new Terraform workspace (or after you add new providers/modules). It does three critical things before plan or apply can succeed:

Downloads and installs all providers referenced in your *.tf files (populating the .terraform/providers/ cache).
Downloads any child modules declared with module blocks.
Initializes the backend (local by default) and creates the lock file .terraform.lock.hcl.

Without it, terraform plan or terraform apply will immediately fail with an error

upvoted 1 times

Question #3 *Topic 3*

As a developer, you want to ensure your plugins are up-to-date with the latest versions.
Which Terraform command should you use?

A. terraform apply -upgrade

B. terraform providers -upgrade

C. terraform init -upgrade

D. terraform refresh -upgrade

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #4 *Topic 3*

What does terraform apply change after you approve the execution plan? (Choose two.)

A. The state file

B. The cloud infrastructure

C. The execution plan

D. The Terraform code

E. The .terraform directory

Suggested Answer: *AB*

Currently there are no comments in this discussion, be the first to comment!

Question #5 *Topic 3*

You modified your local Terraform configuration and ran terraform plan to review the changes. Simultaneously, your teammate manually modified the infrastructure component you are working on.
Since you already ran terraform plan locally, the execution plan for terraform apply will be the same.

A. True

B. False

Suggested Answer: *B*

Srikantha 2 days, 16 hours ago

Selected Answer: B

B. False
The execution plan that terraform apply generates (and then executes) will not be the same as the one you saw from your earlier terraform plan.
Why the plans differ

Refresh happens again
By default, both terraform plan and terraform apply perform a refresh before calculating the diff. This means Terraform queries the real infrastructure (via the provider) to update the in-memory state with current values.
Your teammate's manual change = drift
Between the time you ran terraform plan and when you run terraform apply, the live resource has changed. The refresh during apply will pick up that new reality, so the computed diff (what needs to be created/changed/destroyed) is now different.
upvoted 1 times

Question #6 *Topic 3*

Which is the correct workflow for deploying new infrastructure with Terraform?

A. 1. Write Terraform configuration.
2. Run terraform apply to create infrastructure.
3. Use terraform validate to confirm Terraform deployed resources correctly.

B. 1. Write Terraform configuration.
2. Run terraform init to initialize the working directory or workspace.
3. Run terraform apply.

C. 1. Write Terraform configuration.
2. Run terraform plan to initialize the working directory or workspace.
3. Run terraform apply to create the infrastructure.

D. 1. Write Terraform configuration.
2. Run terraform show to view proposed changes.
3. Run terraform apply to create new infrastructure.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #7 *Topic 3*

After creating a new Terraform configuration, your config passes terraform validate but gives an “Access Denied” error from the cloud provider when running terraform plan.
Why didn’t validate catch this issue?

A. The working directory was not initialized, so the cloud provider plugin wasn’t available to use when running the terraform validate command.

B. The remote backend wasn’t configured, so terraform validate couldn’t load the state and detect the missing credentials.

C. terraform validate only checks if a configuration is syntactically correct and internally consistent, and does not communicate with providers.

D. Variables are only applied and validated during a terraform plan, so validate assumed defaults and returned the success message.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #8 *Topic 3*

Only the user that generated a terraform plan may apply it.

A. True

B. False

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #9 *Topic 3*

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

A. True

B. False

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #10 *Topic 3*

What's the proper syntax for the plan command?

A. terraform plan -generate-config-out=tfplan

B. terraform plan -target=tfplan

C. terraform plan -out=tfplan

D. terraform plan -var-file=tfplan

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #11 *Topic 3*

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

A. It only operates on infrastructure defined in the current working directory or workspace.

B. You cannot target specific resources for the operation.

C. Depending on provider specification, Terraform may need to destroy and recreate your infrastructure resources.

D. You must pass the output of a terraform plan command to it.

E. By default, it does not refresh your state file to reflect the current infrastructure configuration.

Suggested Answer: *AC*

Currently there are no comments in this discussion, be the first to comment!

Question #12 *Topic 3*

Which syntax check returns an error when you run terraform validate?

A. None of these will return an error.

B. There is a missing variable block.

C. The state file does not match the current infrastructure.

D. The code contains tabs for indentation instead of spaces.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #13 *Topic 3*

What is the purpose of the .terraform directory in a Terraform workspace?

A. The directory contains plugins and modules that Terraform downloads during initialization, along with other important information.

B. The directory contains the provide credentials and the .tfvars files to prevent them from being committed to version control by accident.

C. The directory is where Terraform creates and maintains the state file to track the underlying resources it creates and manages.

D. The directory is used to convert and store Terraform configuration files into API calls to communicate with the targeted platform.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #14 *Topic 3*

How can terraform plan aid in the development process?

A. Validates your expectations against the execution plan without permanently modifying state.

B. Reconciles Terraform's state against deploys resources and permanently modifies state using the current status of deployed resources.

C. Initializes your working directory containing your Terraform configuration files.

D. Formats your Terraform configuration files.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #15 *Topic 3*

You have a saved execution plan containing desired changes for infrastructure managed by Terraform. After running the command terraform apply my.tfplan, you receive the error shown in the Exhibit space on this page.

| Error: Saved plan is stale
|
| The given plan file can no longer be applied because the state was changed by another operation after the plan was created.

How can you apply the desired changes? (Choose two.)

A. Generate a new execution plan file with terraform plan, and apply the new plan.

B. Run terraform apply without the saved execution plan.

C. Refresh the current state data using the -refresh-only flag.

D. Force the apply command by adding the flag -lock=false.

E. Update the current plan file using the terraform state push command.

Suggested Answer: *AB*

Currently there are no comments in this discussion, be the first to comment!

Question #16 *Topic 3*

Which of these can you do with terraform plan? (Choose two.)

A. Schedule Terraform to run at a planned time in the future.

B. View the execution plan and check if the changes match your expectations.

C. Execute a plan in a different workspace.

D. Save a generated execution plan to apply later.

Suggested Answer: *BD*

Currently there are no comments in this discussion, be the first to comment!

Question #17 *Topic 3*

Which task does terraform init not perform?

A. Sources any remote modules and downloads them.

B. Connects to the configured backend.

C. Validates that values are set for all required input variables.

D. Sources all providers used in the configuration and downloads them.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #18 *Topic 3*

terraform init retrieves and caches the configuration for all remote modules.

A. True

B. False

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #19 *Topic 3*

You just upgraded the version of a provider in an existing Terraform configuration. What will install the new provider?

A. Run terraform plan.

B. Run terraform apply.

C. Run terraform init -upgrade.

D. Run terraform refresh.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #20 *Topic 3*

terraform validate confirms the syntax of Terraform files.

A. True

B. False

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #21 *Topic 3*

A Terraform configuration is returning errors when running terraform plan.

How would you enable debug logging to investigate further?

A. Rerun terraform plan with the correct flag.

B. Edit the client configuration file.

C. Update the backend configuration.

D. Set the correct environment variable.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #22 *Topic 3*

Why would you use the -replace flag for terraform apply?

A. You want to force Terraform to destroy a resource on the next apply.

B. You want Terraform to ignore a resource on the next apply.

C. You want Terraform to destroy all the infrastructure in your workspace.

D. You want to force Terraform to destroy and recreate a resource on the next apply.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #23 *Topic 3*

How does Terraform determine dependencies between resources when it creates an execution plan?

A. Terraform builds a resource graph based on your configuration and your state file (if present).

B. Terraform requires resources in your configuration be listed in the order they will be created to determine dependencies.

C. Terraform requires all dependencies between resources be specified using the depends_on parameter.

D. Terraform requires resource dependencies be defined as modules and sourced in order.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #1 *Topic 4*

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

A. default

B. value

C. description

D. sensitive

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #2 *Topic 4*

Outside of the required_providers block, Terraform configurations always refer to providers by their local names.

A. True

B. False

Suggested Answer: *A*

Community vote distribution

B (100%)

Srikantha 11 hours, 21 minutes ago

Selected Answer: A

A. True

In Terraform, the required_providers block (inside the terraform {} block) is the only place where you ever reference a provider by its full source address and version constraint. Example:

hclterraform {

required_providers {

aws = {

source = "hashicorp/aws"

version = "~> 5.0"

}

}

}

Everywhere else in the configuration — resource blocks, data sources, provider configuration blocks, provider meta-arguments, modules, etc. — Terraform uses only the local name (the key from required_providers, which defaults to the provider type if the block is omitted)

upvoted 1 times

AZJunkie 3 weeks, 6 days ago

Selected Answer: B

Terraform may use:

The default provider name, or

An alias, if you configured one.

https://developer.hashicorp.com/terraform/language/block/provider

upvoted 1 times

Question #3 *Topic 4*

The Exhibit section of this page shows part of a configuration you've been asked to update. data "azurerm_resource_group" "example" { name = var.resource_group_name
}
resource "azurerm_virtual_network" "example" {
name = _____
}
The name of the Azure Virtual Network should be set to the name of the resource group followed by a dash and the word "vnet".
Which expression fulfills this requirement?

A. "${azurerm_resource_group.example.name}-vnet"

B. join("-",var.resource_group_name,"vnet")

C. concat(data.azurerm_resource_group.example.name, "-", "vnet")

D. "${data.azurerm_resource_group.example.name}-vnet"

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #4 *Topic 4*

Your team is collaborating on infrastructure using Terraform and are formatting their code to follow Terraform language style conventions. How can you update your code to meet these requirements?

A. Run terraform validate prior to executing terraform plan or terraform apply.

B. Replace all tabs with spaces within your Terraform configuration files.

C. Terraform automatically formats configuration on terraform apply.

D. Run terraform fmt to update your Terraform configurations.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #5 *Topic 4*

A resource block is shown in the Exhibit space of this page.
resource “aws_instance” “web” {
count = 2
name = “terraform-${count.index}”
}
How do you reference the name value of the second instance of this resource?

A. aws_instance.web.*.name

B. aws_instance.web[2].name

C. element(aws_instance.web, 2)

D. aws_instance.web[1].name

E. aws_instance.web[1]

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #6 *Topic 4*

When using multiple configurations of the same Terraform provider, what meta-argument must you include in any non-default provider configurations?

A. alias

B. name

C. depends_on

D. id

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #7 *Topic 4*

You want to create a string that is a combination of a generated random_id and a variable and reuse that string several times in your configuration. What is the simplest correct way to implement this without repeating the random_id and variable?

A. Add an output value.

B. Add a local value.

C. Use a data source.

D. Use a module.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #8 *Topic 4*

Your Terraform configuration manages a resource that requires maximum uptime. You need to update the resource, and when you run terraform plan, Terraform notes that the update requires the resource to be destroyed and recreated.
Which lifecycle rule can you add to the resource to reduce the downtime of the resource while still applying the update?

A. prevent_destroy = true

B. destroy = false

C. create_before_destroy = true

D. ignore_changes = all

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #9 *Topic 4*

The Terraform configuration shown in the Exhibit space on this page will create a new AWS instance. data “aws_instance” “web” { filter { name = “tag:Name” values = [“web”]
}
}

A. True

B. False

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #10 *Topic 4*

What kind of configuration block will manage an infrastructure object with settings specified within the block?

A. data

B. locals

C. provider

D. resource

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #11 *Topic 4*

You can execute terraform fmt to standardize all Terraform configurations within the current working directory to Terraform's canonical format and style.

A. True

B. False

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #12 *Topic 4*

A data source is shown in the Exhibit space of this page.

```
data “aws_ami” “web” {
most_recent = true
owners = [“self”]
tags = {
Name = “web-server”
}
}
```

How do you reference the id attribute of this data source?

A. data.aws_ami.web.id

B. web.id

C. aws_ami.web.id

D. data.web.id

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #13 *Topic 4*

A resource block is shown in the Exhibit space of this page.
resource “kubernetes_namespace” “example” {
name = “test”
}
How would you reference the attribute name of this resource in HCL?

A. resource.kubernetes_namespace.example.name

B. data.kubernetes_namespace.name

C. kubernetes_namespace.example.name

D. kubernetes_namespace.test.name

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #14 *Topic 4*

You need to deploy resources into two different regions in the same Terraform configuration using the block in the Exhibit space on this page.

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

What do you need to add to the provider configuration to deploy the resource to the us-west-2 AWS region?

A.
```
resource “aws_instance” “example-us-west-2” {
ami = data.aws_ami.ubuntu.id
instance_type = “t3.micro”
}
```

B.
```
provider “aws” {
region = “us-east-1”
}
provider “aws” “west” {
region = “us-west-2”
}
```

C.
```
provider “aws_west” {
region = “us-west-2”
}
```

D.
```
provider “aws” {
region = “us-east-1”
}
provider “aws” {
alias = “west”
region = “us-west-2”
}
```

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!