CertyIQ

CertyIQ

Premium exam material

Get certification quickly with the CertyIQ Premium exam material.
Everything you need to prepare, learn & pass your certification exam easily. Lifetime free updates
First attempt guaranteed success.

https://www.CertyIQ.com

About CertyIQ

We here at CertyIQ eventually got enough of the industry's greedy exam paid for. Our team of IT professionals comes with years of experience in the IT industry Prior to training CertIQ we worked in test areas where we observed the horrors of the paywall exam preparation system.

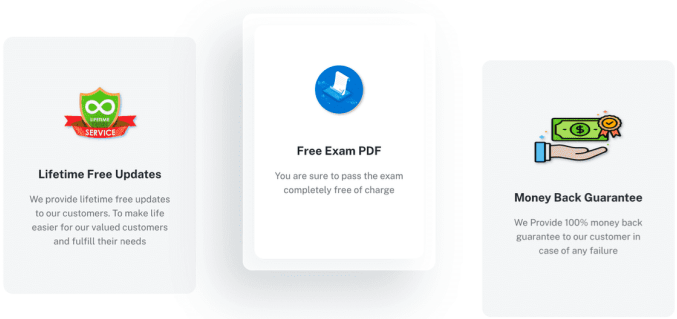
The misuse of the preparation system has left our team disillusioned. And for that reason, we decided it was time to make a difference. We had to make In this way, CertyIQ was created to provide quality materials without stealing from everyday people who are trying to make a living.

Doubt Support

We have developed a very scalable solution using which we are able to solve 400+ doubts every single day with an average rating of 4.8 out of 5.

https://www.certyiq.com

Mail us on - certyiqofficial@gmail.com



John

October 19, 2022

★★★★★

Thanks you so much for your help. I scored 972 in my exam today. More than 90% were from your PDFs!

Dana

September 04, 2022

★★★★★

Thanks a lot for this updated AZ-900 Q&A. I just passed my exam and got 974, I followed both of your Az-900 videos and the 6 PDF, the PDFs are very much valid, all answers are correct. Could you please create a similar video/PDF for DP900, your content/PDF's is really awesome. The team did a really good job. Thank You 😊.

Ahamed Shibly

2 months ago

★★★★★

Customer support is realy fast and helpful, I just finished my exam and this video along with the 6 PDF helped me pass! Definitely recommend getting the PDFs. Thank you!

October 22, 2022

★★★★★

Passed my exam today with 891 marks. Out of 52 questions, 51 were from certyiq PDFs including Contoso case study.
Thank You certyiq team!

Henry Rome

2 months ago

★★★★★

These questions are real and 100 % valid. Thank you so much for your efforts, also your 4 PDFs are awesome, I passed the DP900 exam on 1 Sept. With 968 marks. Thanks a lot, buddy!

Esmaria

2 months ago

★★★★★

Simple easy to understand explanations. To anyone out there wanting to write AZ900, I highly recommend 6 PDF's.Thank you so much, appreciate all your hard work in having such great content. Passed my exam Today - 3 September with 942 score.

Hashicorp

(Terraform Associate)

HashiCorp Certified

Total: **283 Questions**

Link: https://certyiq.com/papers?provider=hashicorp&exam=terraform-associate

Question: 1 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/language/state/index.html
" target="_blank" style="word-break: break-all;">

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.

Terraform uses this local state to create plans and make changes to your infrastructure. Prior to any operation, Terraform does a refresh to update the state with the real infrastructure.

The primary purpose of Terraform state is to store bindings between objects in a remote system and resource instances declared in your configuration. When Terraform creates a remote object in response to a change of configuration, it will record the identity of that remote object against a particular resource instance, and then potentially update or delete that object in response to future configuration changes.

Question: 2 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

The remote backend can work with either a single remote Terraform Cloud workspace, or with multiple similarly-named remote workspaces (like networking-dev and networking-prod). The workspaces block of the backend configuration determines which mode it uses:

To use a single remote Terraform Cloud workspace, set workspaces.name to the remote workspace's full name (like networking-prod).

To use multiple remote workspaces, set workspaces.prefix to a prefix used in all of the desired remote workspace names. For example, set prefix = "networking-" to use Terraform cloud workspaces with names like networking-dev and networking-prod. This is helpful when mapping multiple Terraform CLI workspaces used in a single Terraform configuration to multiple Terraform Cloud workspaces.

https://developer.hashicorp.com/terraform/language/settings/backends/remote

Question: 3 CertyIQ

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:

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

Question: 4 CertyIQ

What is the workflow for deploying new infrastructure with Terraform?

A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure.

B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.

C. terraform import to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure.

D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

Answer: D

Explanation:

Correct answer is D: Write a Terraform configuration, run terraform in it, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

Question: 5 CertyIQ

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

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

A. True

B. False

Answer: A

Explanation:

Reference:
https://github.com/hashicorp/terraform/issues/17928

Question: 6 CertyIQ

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:

You are all correct that taint has been deprecated and replaced with -replace. But neither D nor any other option here uses the -replace command. Therefore option A is the only valid option given these choices.

Question: 7 CertyIQ

Which provisioner invokes a process on the resource created by Terraform?

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

Answer: A

Explanation:
The remote-exec provisioner invokes a script on a remote resource after it is created.

Reference:
https://www.terraform.io/docs/language/resources/provisioners/remote-exec.html

Question: 8 CertyIQ

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:

E. None of the above

All of the statements are true of Terraform providers.

A. Providers can be written by individuals - Any person or organization can develop and distribute a Terraform provider, allowing them to expand Terra form's capabilities to manage resources that it previously could not.

B. Providers can be maintained by a community of users - Many Terraform providers are open source projects, and the development and maintenance of these providers can be collaborative efforts between multiple individuals and organizations.

C. Some providers are maintained by Hashi Corp - Hashi Corp, the creators of Terraform, maintain a number of official providers that cover popular infrastructure providers such as AWS, Google Cloud, and Microsoft Azure.

D. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers - Providers can be developed and maintained by cloud vendors, non-cloud vendors, or a combination of both, to expand Terraform 's capabilities and support for different types of infrastructure.

Question: 9 CertyIQ

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

Question: 10 CertyIQ

You have deployed a new webapp with a public IP address on a cloud provider. However, you did not create any outputs for your code.
What is the best method to quickly find the IP address of the resource you deployed?

A. Run terraform output ip_address to view the result

B. In a new folder, use the terraform_remote_state data source to load in the state file, then write an output for each resource that you find the state file

C. Run terraform state list to find the name of the resource, then terraform state show to find the attributes

including public IP address

D. Run terraform destroy then terraform apply and look for the IP address in stdout

Answer: C

Explanation:

You can find the info in the state. not A because you don't have to outputs defined.

Question: 11 CertyIQ

Which of the following is not a key principle of infrastructure as code?

A. Versioned infrastructure

B. Golden images

C. Idempotence

D. Self-describing infrastructure

Answer: B

Explanation:

1. it not possible create golden image from terraform, thats, is an other tools from hashicorp
2. B. Golden images"Golden images" refers to a specific method of deployment where a pre-configured image of an operating system or application is stored and used as a base for all new deployments. This method is not a key principle of Infrastructure as Code, but it is a common method of deployment in traditional IT environments.Infrastructure as Code is based on the following key principles:A. Versioned infrastructure - The infrastructure is treated as code and is version controlled, allowing for auditing, rollback, and collaboration.C. Idempotence - The infrastructure is provisioned in a repeatable and predictable manner, making it possible to run the provisioning scripts multiple times without creating additional resources or causing changes to existing resources.D. Self-describing infrastructure - The code used to provision the infrastructure is human-readable and self-documented, making it easier to understand and maintain over time.

Question: 12 CertyIQ

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

A. True

B. False

Answer: B

Explanation:

1. Answer is B. Descriptions aren't stored in the state file.
2. The answer is B. Descriptions aren't stored in the state file.

Question: 13 CertyIQ

What is the provider for this fictitious resource?

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

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

Answer: C

Explanation:

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

Question: 14 CertyIQ

If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?

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

Answer: A

Explanation:

A. Run terraform refresh

Yes, in older versions of Terraform (before 0.15), terraform refresh could be used to reconcile the state file with the actual infrastructure. From Terraform 0.15 and later, terraform apply performs this reconciliation.

B. It will happen automatically

No, Terraform does not automatically detect changes made outside of its context. You must run terraform apply (or terraform plan in older versions) to see these changes and apply them to the Terraform state file.

Question: 15 CertyIQ

What is not processed when running a terraform refresh?

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

Answer: B

Explanation:
1. A. State file - its updated during the refresh B. Configuration file - C. Credentials - required to get onto the cloud D. Cloud provider - required to carry out the refresh of whats in the cloud vs whats in state.
2. The answer is B.

Question: 16 CertyIQ

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

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

Answer: D

Explanation:

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

definitely D since it's 'public' (all input & output variables should be revealed)

Question: 17 CertyIQ

If a module uses a local values, 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

Question: 18 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

It is generally considered insecure to store secret data, such as passwords, API keys, and other sensitive information, in the same version control repository as your Terraform configuration. This is because version control repositories are often publicly accessible, and if sensitive information is stored in the repository it can be easily accessed by unauthorized individuals. Additionally, version control repositories typically have a history of all changes made to files, so even if sensitive information is deleted at a later point, it can still be retrieved from the repository history. To properly secure secret data, it is recommended to store it in a secure and encrypted format, such as in a secure vault or by using a tool specifically designed for storing secrets.

Reference:

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

Question: 19 CertyIQ

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

A. split
B. join
C. slice
D. chomp

Answer: C

Explanation:

slice is not in string function list.

C - Slice that s used for list, not string, chomp is for used in string

Question: 20 CertyIQ

You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with
Terraform and want to manage these VMs using Terraform instead.
What are the two things you must do to achieve this? (Choose two.)

A. Provision new VMs using Terraform with the same VM names
B. Use the terraform import command for the existing VMs
C. Write Terraform configuration for the existing VMs
D. Run the terraform import-gcp command

Answer: BC

Explanation:
1. You should create the equivalent configuration first, and then run import to load it on the state file.
2. To manage the VMs that were provisioned using the gcloud command line tool using Terraform, you must first use the "terraform import" command to import the existing VMs into Terraform's state file. This allows Terraform to recognize the VMs and manage them as part of your infrastructure.After importing the VMs, you must then write Terraform configuration for the existing VMs. This includes defining the resources for the

VMs and specifying the necessary configuration options, such as the instance type and image, as well as any other desired settings. By doing this, you can manage the VMs using Terraform, which provides a single, unified tool for provisioning, configuring, and managing your infrastructure.

Question: 21 CertyIQ

You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season. Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.
How would you solve this using infrastructure as code?

A. Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource

B. Implement a checklist that engineers can follow when configuring compute instances

C. Replace the compute instance type with a larger version to reduce the number of required deployments

D. Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews

Answer: D

Explanation:

Using infrastructure as code (IAC) can help solve the problem of inconsistent configurations by automating the deployment and configuration of compute instances. With IAC, you can define your infrastructure as code, commit it to version control, and then use a provisioning pipeline to automatically deploy and configure the infrastructure based on the code in version control. This ensures that every compute instance is deployed and configured consistently, eliminating the risk of manual configuration errors. Additionally, implementing a provisioning pipeline with code reviews can help catch any potential issues before they are deployed, further reducing the risk of outages during peak shopping time.

Question: 22 CertyIQ

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

A. True

B. False

Answer: B

Explanation:

It's not a must for the file name to be "main.tf". It checks all .t f files in the current directory and does the needful

Tested. no main/sample file after in it

Question: 23 CertyIQ

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

A. Destroy

B. Apply
C. Import
D. Init
E. Validate

Answer: BD

Explanation:

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

Question: 24 CertyIQ

Why would you use the terraform taint command?

A. When you want to force Terraform to destroy a resource on the next apply
B. When you want to force Terraform to destroy and recreate a resource on the next apply
C. When you want Terraform to ignore a resource on the next apply
D. When you want Terraform to destroy all the infrastructure in your workspace

Answer: B

Explanation:

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

Reference:

https://www.terraform.io/docs/cli/commands/taint.html

Question: 25 CertyIQ

Terraform requires the Go runtime as a prerequisite for installation.

A. True
B. False

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/extend/guides/v1-upgrade-guide.html
" target="_blank" style="word-break: break-all;">

As of September 2019, Terraform provider developers importing the Go module `github.com/hashicorp/terraform`, known as Terraform Core, should switch to `github.com/hashicorp/terraform-plugin-sdk`, the Terraform Plugin SDK, instead.

Why a separate module?

While the `helper/*` and other packages in Terraform Core has served us well, in order for provider development to evolve, the SDK needed to break out into its own repository. Terraform Core's versioning has been oriented towards practitioners. With the "unofficial" SDK existing in the core repository, the SDK becomes tied to Core releases and cannot follow semantic versioning. The new standalone SDK github.com/hashicorp/terraform-plugin-sdk follows sematic versioning starting with v1.0.0.

We will use the term "legacy Terraform plugin SDK" when referring to the version of Terraform Core imported and used by providers.

Question: 26 CertyIQ

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. You apply failed due to a state lock

Answer: C

Explanation:
Manually unlock the state for the defined configuration.

Reference:
https://www.terraform.io/docs/cli/commands/force-unlock.html

Question: 27 CertyIQ

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:

Correct answer is A:FTP server

Question: 28 CertyIQ

Which of the following is available only in Terraform Enterprise or Cloud workspaces and not in Terraform CLI?

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

Answer: A

Explanation:

Correct answer is A:Secure variable storage

Question: 29 CertyIQ

terraform validate validates the syntax of Terraform files.

A. True
B. False

Answer: A

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

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

Question: 30 CertyIQ

You have used Terraform to create an ephemeral development environment in the cloud and are now ready to destroy all the infrastructure described by your
Terraform configuration. To be safe, you would like to first see all the infrastructure that will be deleted by Terraform.
Which command should you use to show all of the resources that will be deleted? (Choose two.)

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

Answer: AD

Explanation:

A. Run terraform plan -destroy.

D. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval.

Question: 31 CertyIQ

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:

D to pass a value to a module the syntax is <child module variable name > = <value>Here child module variable name is server reference the value of a variable in main/root module var. <name _of _the variable>

Question: 32 CertyIQ

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

Question: 33 CertyIQ

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:

B is correct answer : false.

Question: 34 CertyIQ

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

Question: 35 CertyIQ

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:

Correct answer is A:Run the terraform f m t command during the code linting phase of your CI/CD process

Question: 36 CertyIQ

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.

Question: 37 CertyIQ

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
" target="_blank" style="word-break: break-all;">

Usage

Usage: `terraform init [options]`

This command performs several different initialization steps in order to prepare the current working directory for use with Terraform. More details on these are in the sections below, but in most cases it is not necessary to worry about these individual steps.

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.

Question: 38 CertyIQ

You have declared a variable called var.list which is a list of objects that all have an attribute id.
Which options will produce a list of the IDs? (Choose two.)

A. for o in var.list : o => o.id
B. var.list[*].id
C. [var.list[*].id]
D. [for o in var.list : o.id]

Answer: BD

Explanation:

https://www.terraform.io/language/expressions/splatA splat expression provides a more concise way to express a common operation that could otherwise be performed with a for expression.

Question: 39 CertyIQ

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: E

Explanation:
1. The type argument in a variable block allows you to restrict the type of value that will be accepted as the value for a variable. If no type constraint is set then a value of any type is accepted.Source: https://www.terraform.io/language/values/variables
2. non of the options is mandatory - they are all optional

Question: 40 CertyIQ

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:

C is correct answer.

The version argument accepts a version constraint string. Terraform will use the newest installed version of the module that meets the constraint; if no acceptable versions are installed, it will download the newest version that meets the constraint.

Version constraints are supported only for modules installed from a module registry, such as the public Terraform Registry or Terraform Cloud's private module registry. Other module sources can provide their own versioning mechanisms within the source string itself, or might not support versions at all. In particular, modules sourced from local file paths do not support version; since they're loaded from the same source repository, they always share the same version as their caller.

Reference:

https://www.terraform.io/language/modules/syntax#version

Question: 41 CertyIQ

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:

C and D are correct answers, there is no auto backup in Terraform cloud.

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

Question: 42 CertyIQ

Where does the Terraform local backend store its state?

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

Answer: C

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

Question: 43 CertyIQ

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

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

Answer: D

Explanation:

1. It's DWe can use providers to supply variable values (vault for example).We can provide input variable value in parameter for apply command.We can use environment variables.HashiCorp is not mentioning anything about secure strings.Reference:https://www.terraform.io/language/values/variables
2. D. secure string is not an option for keeping secrets out of Terraform configuration files.Secure string: While there is no option for a "secure string" in Terraform, you can use a number of different techniques to encrypt or obfuscate sensitive information in your configuration files. For example, you might use a tool like SOPS to encrypt your Terraform code, or you might use a tool like Vault to store and manage your secrets separately from your code.

Question: 44 CertyIQ

What is one disadvantage of using dynamic blocks in Terraform?

A. They cannot be used to loop through a list of values
B. Dynamic blocks can construct repeatable nested blocks
C. They make configuration harder to read and understand
D. Terraform will run more slowly

Answer: C

Explanation:

1. Looking at documentation "Overuse of dynamic blocks can make configuration hard to read and maintain"
2. correct answer

Question: 45 CertyIQ

Only the user that generated a plan may apply it.

A. True
B. False

Answer: B

Explanation:

B is correct, a plan can be stored as a file and another person can execute the plan file

Question: 46 CertyIQ

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

Question: 47 CertyIQ

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.

Answer: depends_on

Question: 48 CertyIQ

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. Execute terraform destroy and select the newly-created VM.

B. The Terraform state file only contains the one new VM. Execute terraform destroy.

C. Delete the Terraform state file and execute Terraform apply.

D. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.

Answer: B

Explanation:

Is B for sure. the state file does not contain information about anything else that you might have on the cloud provider.

Question: 49 CertyIQ

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

```
resource "azurerm_resource_group" "dev" {
  name = "test"
  location = "westus"
}
```

A. dev

B. azurerm_resource_group

C. azurerm

D. test

Answer: A

Explanation:

Correct answer is A:dev

Question: 50 CertyIQ

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

A. True

B. False

Answer: B

Explanation:

1. TF_LOG will enable log and send that to stderr (by default it i screen) not to any syslog server. TF_LOG_PARH can be set to redirect logs to a specific file

2. B is correct answer. Use TF_LOG_PATH to specify a file path where the log output file should be written. If you do not specify a log path, Terraform writes the specified log output to stderr." Stderr, also known as

standard error, is the default file descriptor where a process can write error messages. In Unix-like operating systems, such as Linux, macOS X, and BSD, stderr is defined by the POSIX standard. Its default file descriptor number is 2.In the terminal, standard error defaults to the user's screen."

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

20standard,defaults%20to%20the%20user's%20screen.

Question: 51 CertyIQ

Where in your Terraform configuration do you specify a state backend?

A. The terraform block
B. The resource block
C. The provider block
D. The datasource block

Answer: A

Explanation:
Backends are configured with a nested backend block within the top-level terraform block.

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

Question: 52 CertyIQ

In Terraform 0.13 and above, outside of the required_providers block, Terraform configurations always refer to providers by their local names.

A. True
B. False

Answer: A

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

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

Question: 53 CertyIQ

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

Question: 54 CertyIQ

Terraform providers are always installed from the Internet.

A. True
B. False

Answer: B

Explanation:
Terraform configurations must declare which providers they require, so that Terraform can install and use them.

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

Question: 55 CertyIQ

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

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

Answer: C

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

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

Question: 56 CertyIQ

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

Explanation:

f you are creating a new virtual machine using Terraform, it may take a few minutes for the virtual machine to be created and for it to become available for use. During this time, Terraform will continue to report on the progress of the creation process and will display any errors or issues that may arise. Once the virtual machine has been successfully created, the changes will be reflected in your cloud environment. I go with B

Question: 57 CertyIQ

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:

B - aws_instance.web[1].name !!!

Index starts at 0 so the second instance would be 1 - the link below confirms this:

https://www.terraform.io/language/meta-arguments/count#referring-to-instances

Question: 58 CertyIQ

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:

1. The answer should be BA terraform can only provision resouce in one Cloud not multiple cloud
2. "A" Terraform provider is responsible for "A" Cloud."Multiple" Terraform providers can be responsible for "Multiple" Clouds.

Question: 59 CertyIQ

Terraform provisioners can be added to any resource block.

A. True
B. False

Answer: A

Explanation:

Reference:
https://www.terraform.io/docs/language/resources/provisioners/syntax.html
" target="_blank" style="word-break: break-all;">

Terraform includes the concept of provisioners as a measure of pragmatism, knowing that there will always be certain behaviors that can't be directly represented in Terraform's declarative model.

However, they also add a considerable amount of complexity and uncertainty to Terraform usage. Firstly, Terraform cannot model the actions of provisioners as part of a plan because they can in principle take any action. Secondly, successful use of provisioners requires coordinating many more details than Terraform usage usually requires: direct network access to your servers, issuing Terraform credentials to log in, making sure that all of the necessary external software is installed, etc.

The following sections describe some situations which can be solved with provisioners in principle, but where better solutions are also available. We do not recommend using provisioners for any of the use-cases described in the following sections.

Even if your specific use-case is not described in the following sections, we still recommend attempting to solve it using other techniques first, and use provisioners only if there is no other option.

Question: 60 CertyIQ

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:

Reference:
https://www.hashicorp.com/blog/detecting-and-managing-drift-with-terraform
" target="_blank" style="word-break: break-all;">

Prior to a plan or apply operation, Terraform does a refresh to update the state file with real-world status. You can also do a refresh any time with `terraform refresh`:

```
$ terraform refresh
aws_instance.example: Refreshing state... (ID: i-011a9893eff09ede1)
```

What Terraform is doing here is reconciling the resources tracked by the state file with the real world. It does this by querying your infrastructure providers to find out what's actually running and the current configuration, and updating the state file with this new information. Terraform is designed to co-exist with other tools as well as manually provisioned resources and so it only refreshes resources under its management.

The output for a refresh is minimal. Terraform lists each resource it is refreshing along with its internal ID. Running `refresh` does not modify infrastructure, but does modify the state file. If the state has drifted from the last time Terraform ran, `refresh` allows that drift to be detected.

Question: 61 CertyIQ

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.

Answer: -out=FILENAME

Explanation:

Reference:
https://www.terraform.io/docs/cli/commands/plan.html
" target="_blank" style="word-break: break-all;">

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.

Question: 62 CertyIQ

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.

Answer:

Terraform.tfstate

Explanation:

Reference:
https://www.terraform.io/docs/language/state/index.html
" target="_blank" style="word-break: break-all;">

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.

Question: 63 CertyIQ

A Terraform local value can reference other Terraform local values.

A. True
B. False

Answer: A

Explanation:

Reference:
https://www.terraform.io/docs/configuration-0-11/locals.html
" target="_blank" style="word-break: break-all;">

The `locals` block defines one or more local variables within a module. Each `locals` block can have as many locals as needed, and there can be any number of `locals` blocks within a module.

The names given for the items in the `locals` block must be unique throughout a module. The given value can be any expression that is valid within the current module.

The expression of a local value can refer to other locals, but as usual reference cycles are not allowed. That is, a local cannot refer to itself or to a variable that refers (directly or indirectly) back to it.

It's recommended to group together logically-related local values into a single block, particularly if they depend on each other. This will help the reader understand the relationships between variables. Conversely, prefer to define *unrelated* local values in *separate* blocks, and consider annotating each block with a comment describing any context common to all of the enclosed locals.

Question: 64 CertyIQ

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

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

Answer: C

Explanation:

Reference:
https://www.terraform.io/docs/language/expressions/type-constraints.html
" target="_blank" style="word-break: break-all;">

The three kinds of collection type in the Terraform language are:

- `list(...)` : a sequence of values identified by consecutive whole numbers starting with zero.

  The keyword `list` is a shorthand for `list(any)` , which accepts any element type as long as every element is the same type. This is for compatibility with older configurations; for new code, we recommend using the full form.

- `map(...)` : a collection of values where each is identified by a string label.

  The keyword `map` is a shorthand for `map(any)` , which accepts any element type as long as every element is the same type. This is for compatibility with older configurations; for new code, we recommend using the full form.

  Maps can be made with braces ({}) and colons (:) or equals signs (=): { "foo": "bar", "bar": "baz" } OR { foo = "bar", bar = "baz" }. Quotes may be omitted on keys, unless the key starts with a number, in which case quotes are required. Commas are required between key/value pairs for single line maps. A newline between key/value pairs is sufficient in multi-line maps.

  Note: although colons are valid delimiters between keys and values, they are currently ignored by `terraform fmt` (whereas `terraform fmt` will attempt vertically align equals signs).

- `set(...)` : a collection of unique values that do not have any secondary identifiers or ordering.

Question: 65 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

its not immediately

The terraform taint command informs Terraform that a particular object has become degraded or damaged. Terraform represents this by marking the object as "tainted" in the Terraform state, and Terraform will propose to replace it in the next plan you create.

Reference:

https://www.devopsschool.com/blog/terraform-taint-and-untaint-explained-with-example-programs-and-tutorials/

Question: 66 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

1. not all of them so False

2. B is correct answer : False. "By default, Terraform uses a backend called local, which stores state as a local file on disk. You can also configure one of the built-in backends listed in the documentation sidebar.Some of these backends act like plain remote disks for state files, while others support locking the state while operations are being performed. This helps prevent conflicts and inconsistencies. The built-in backends listed are the only backends. You cannot load additional backends as plugins.

"https://www.terraform.io/language/settings/backends/configuration#available-backends

Question: 67 CertyIQ

How can terraform plan aid in the development process?

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

B. Initializes your working directory containing your Terraform configuration files

C. Formats your Terraform configuration files

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

Answer: A

Explanation:

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

Terra form's plan command is used to create an execution plan that outlines the steps that Terraform will take to reach your desired infrastructure state. It allows you to preview and validate the changes that will be made to your infrastructure before actually making those changes. This can be helpful in the development process because it allows you to see exactly what will be changed and ensure that it aligns with your expectations before you apply those changes.

Reference:

https://github.com/hashicorp/terraform/issues/19235

Question: 68 CertyIQ

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:

B is correct answer : Workspaces.

https://www.terraform.io/language/state/workspaces#when-to-use-multiple-workspaces

Question: 69 CertyIQ

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. teat

Answer: B

Explanation:

Main

is the name of the resources

Question: 70 CertyIQ

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:

Reference:

https://developer.hashicorp.com/terraform/language/values/variables

Question: 71 CertyIQ

Your security team scanned some Terraform workspaces and found secrets stored in a plaintext in state files.
How can you protect sensitive data stored in Terraform state files?

A. Delete the state file every time you run Terraform

B. Store the state in an encrypted backend

C. Edit your state file to scrub out the sensitive data

D. Always store your secrets in a secrets.tfvars file.

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/language/state/sensitive-data.html
" target="_blank" style="word-break: break-all;">

Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

For example:

- Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.
- The S3 backend supports encryption at rest when the `encrypt` option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

Question: 72 CertyIQ

In contrast to Terraform Open Source, when working with Terraform Enterprise and Cloud Workspaces, conceptually you could think about them as completely separate working directories.

A. True

B. False

Answer: A

Explanation:

Answer is A

From the doc: "Terraform Cloud manages infrastructure collections with workspaces instead of directories. A workspace contains everything Terraform needs to manage a given collection of infrastructure, and separate workspaces function like completely separate working directories."

https://www.terraform.io/cloud-docs/workspaces

Question: 73 CertyIQ

You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (*.tf 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:

Reference:
https://www.terraform.io/docs/cli/config/environment-variables.html
" target="_blank" style="word-break: break-all;">

Terraform refers to a number of environment variables to customize various aspects of its behavior. None of these environment variables are required when using Terraform, but they can be used to change some of Terraform's default behaviors in unusual situations, or to increase output verbosity for debugging.

TF_LOG

Enables detailed logs to appear on stderr which is useful for debugging. For example:

```
export TF_LOG=trace
```
Copy

Question: 74 CertyIQ

How is terraform import run?

A. As a part of terraform init
B. As a part of terraform plan
C. As a part of terraform refresh
D. By an explicit call

E. All of the above

Answer: D

Explanation:

Correct answer is D:By an explicit call

Question: 75 CertyIQ

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

Explanation:

Correct answer is D. Terraform will recreate the VM.

In Terraform, the state file is used to store the current state of your infrastructure. When you run terraform apply, Terraform compares the state of your infrastructure as defined in the configuration files with the state recorded in the state file, and then makes any necessary changes to bring the infrastructure into compliance with the configuration.

Question: 76 CertyIQ

Which of these options is the most secure place to store secrets foe 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:

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.

Reference:

https://www.terraform.io/language/settings/backends/configuration#credentials-and-sensitive-data

Question: 77 CertyIQ

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

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

Answer: C

Explanation:

Reference:

https://www.terraform.io/cdktf/concepts/remote-backends
https://developer.hashicorp.com/terraform/cdktf/concepts/remote-backends

Question: 78 CertyIQ

Which backend does the Terraform CLI use by default?

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

Answer: D

Explanation:

D is correct answer : local

"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."

Reference:

https://www.terraform.io/docs/language/settings/backends/configuration.html

Default Backend

If a configuration includes no backend block, Terraform defaults to using the `local` backend, which stores state as a plain file in the current working directory.

Question: 79 CertyIQ

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:

C is correct answer : hidden terraform directory

"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."

Reference:

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

Question: 80 CertyIQ

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

Explanation:

1. Need to run Terraform Init first to install the plugins
2. B is the correct, terraform init to download required plugins

Question: 81 CertyIQ

What features stops multiple admins from changing the Terraform state at the same time?

A. Version control
B. Backend types
C. Provider constraints
D. State locking

Answer: D

Explanation:

Reference:
https://blog.gruntwork.io/how-to-manage-terraform-state-28f5697e68fa
" target="_blank" style="word-break: break-all;">

2. **Locking**: Most version control systems do not provide any form of locking that would prevent two team members from running `terraform apply` on the same state file at the same time.

Question: 82 CertyIQ

A fellow developer on your team is asking for some help in refactoring their Terraform code. As part of their application's architecture, they are going to tear down an existing deployment managed by Terraform and deploy new. However, there is a server resource named aws_instance.ubuntu[1] they would like to keep to perform some additional analysis.
What command should be used to tell Terraform to no longer manage the resource?

A. terraform apply rm aws_instance.ubuntu[1]
B. terraform state rm aws_instance.ubuntu[1]
C. terraform plan rm aws_instance.ubuntu[1]
D. terraform delete aws_instance.ubuntu[1]

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/cli/commands/state/rm.html
" target="_blank" style="word-break: break-all;">

Usage

Usage: `terraform state rm [options] ADDRESS...`

Terraform will search the state for any instances matching the given resource address, and remove the record of each one so that Terraform will no longer be tracking the corresponding remote objects.

This means that although the objects will still continue to exist in the remote system, a subsequent `terraform plan` will include an action to create a new object for each of the "forgotten" instances. Depending on the constraints imposed by the remote system, creating those objects might fail if their names or other identifiers conflict with the old objects still present.

Question: 83 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

B. False.

Terraform can manage resource dependencies automatically without explicitly setting them using the depends_on argument.

Question: 84 CertyIQ

A terraform apply can not _____ infrastructure.

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

Answer: D

Explanation:

Correct answer is D: import

Question: 85 CertyIQ

You need to constrain the GitHub provider to version 2.1 or greater.
Which of the following should you put into the Terraform 0.12 configuration's provider block?

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

Answer: B

Explanation:

Reference:
https://github.com/hashicorp/terraform-provider-null/issues/31

Question: 86 CertyIQ

You just scaled your VM infrastructure and realized you set the count variable to the wrong value. You correct the value and save your change.
What do you do next to make your infrastructure match your configuration?

A. Run an apply and confirm the planned changes
B. Inspect your Terraform state because you want to change it
C. Reinitialize because your configuration has changed
D. Inspect all Terraform outputs to make sure they are correct

Answer: A

Explanation:

A is correct answer : apply

You need your deployment to match your config, so only way to implement changes is through terraform apply.

Question: 87 CertyIQ

Terraform provisioners that require authentication can use the _____ block.

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

Answer: A

Explanation:

The answer is A. connection. The connection block is used to configure the authentication settings for a Terraform provisioner. This includes the username, password, and SSH key that will be used to connect to the remote resource. The credentials and secrets blocks are used to store sensitive information, such as passwords and SSH keys. These blocks are not directly used by provisio ners, but they can be referenced by the connection block. The ssh block is used to configure the SSH client that Terraform will use to connect to the remote resource. This block is not typically used by provisioners, as the connection block can be used to configure the SSH client for all provisioners. Here is an example of a connection block for a provisioner that requires SSH authentication:connection host = "example.com" user = "root" password = "my-password"

https://developer.hashicorp.com/terraform/cli/config/config-file

Question: 88 CertyIQ

Terraform validate reports syntax check errors from which of the following scenarios?

A.Code contains tabs indentation instead of spaces

B.There is missing value for a variable

C.The state files does not match the current infrastructure

D.None of the above

Answer: B

Explanation:

B is correct answer. The terraform validate command is used to validate the syntax of the terraform files. Terraform performs a syntax check on all the terraform files in the directory, and will display an error if any of the files doesn't validate.This command does not check formatting (e.g. tabs vs spaces, newlines, comments etc.).The following can be reported:invalid HCL syntax (e.g. missing trailing quote or equal sign)invalid HCL references (e.g. variable name or attribute which doesn't exist)same provider declared multiple timessame module declared multiple timessame resource declared multiple timesinvalid module nameinterpolation used in places where it's unsupported (e.g. variable, depends_on, module.source, provider)missing value for a variable (none of -var foo=... flag, -var-file=foo.vars flag, TF_VAR_foo environment variable, terraform.tfvars, or default value in the configuration)

Reference:

http://man.hubwiz.com/docset/Terraform.docset/Contents/Resources/Documents/docs/commands/validate.html

The `terraform validate` command is used to validate the syntax of the terraform files. Terraform performs a syntax check on all the terraform files in the directory, and will display an error if any of the files doesn't validate.

This command **does not** check formatting (e.g. tabs vs spaces, newlines, comments etc.).

The following can be reported:

- invalid HCL syntax (e.g. missing trailing quote or equal sign)
- invalid HCL references (e.g. variable name or attribute which doesn't exist)
- same `provider` declared multiple times
- same `module` declared multiple times
- same `resource` declared multiple times
- invalid `module` name
- interpolation used in places where it's unsupported (e.g. `variable`, `depends_on`, `module.source`, `provider`)
- missing value for a variable (none of `-var foo=...` flag, `-var-file=foo.vars` flag, `TF_VAR_foo` environment variable, `terraform.tfvars`, or default value in the configuration)

Question: 89 CertyIQ

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

A.count
B.name
C.source
D.version

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/language/values/variables.html
" target="_blank" style="word-break: break-all;">

The label after the `variable` keyword is a name for the variable, which must be unique among all variables in the same module. This name is used to assign a value to the variable from outside and to reference the variable's value from within the module.

The name of a variable can be any valid identifier *except* the following: `source`, `version`, `providers`, `count`, `for_each`, `lifecycle`, `depends_on`, `locals`.

These names are reserved for meta-arguments in module configuration blocks, and cannot be declared as variable names.

Question: 90 CertyIQ

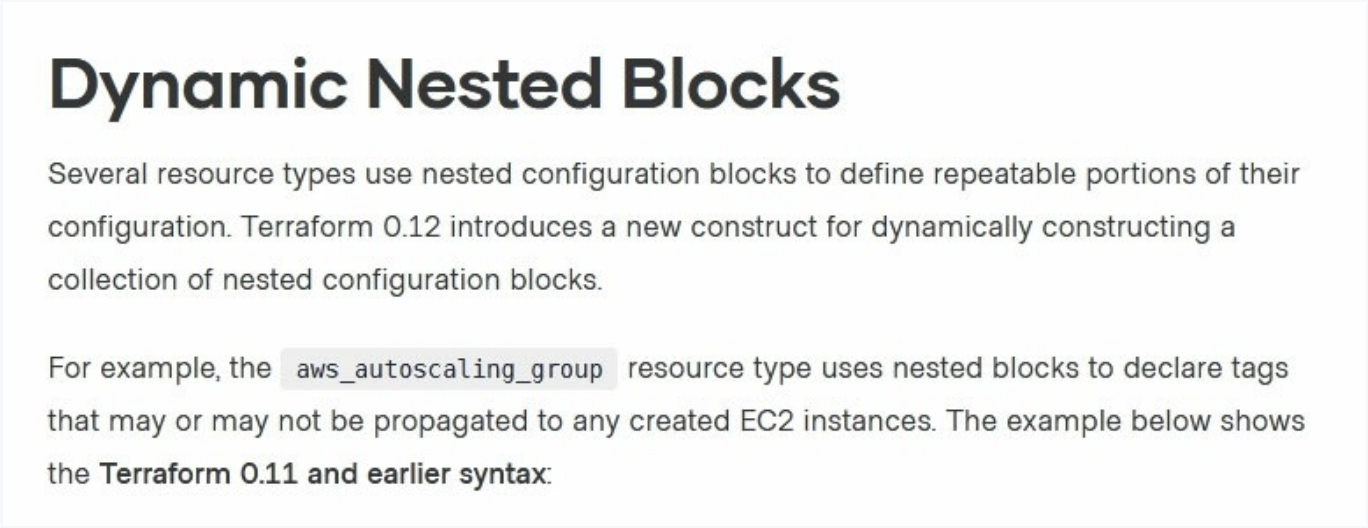
What type of block is used to construct a collection of nested configuration blocks?

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

Answer: D

Explanation:

Reference:
https://www.hashicorp.com/blog/hashicorp-terraform-0-12-preview-for-and-for-each
" target="_blank" style="word-break: break-all;">



Dynamic Nested Blocks

Several resource types use nested configuration blocks to define repeatable portions of their configuration. Terraform 0.12 introduces a new construct for dynamically constructing a collection of nested configuration blocks.

For example, the `aws_autoscaling_group` resource type uses nested blocks to declare tags that may or may not be propagated to any created EC2 instances. The example below shows the **Terraform 0.11 and earlier syntax**:

Question: 91 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

Reference:
https://github.com/hashicorp/terraform/issues/15818

Question: 92 CertyIQ

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:

Reference:
https://www.terraform.io/docs/language/syntax/style.html
" target="_blank" style="word-break: break-all;">

The Terraform parser allows you some flexibility in how you lay out the elements in your configuration files, but the Terraform language also has some idiomatic style conventions which we recommend users always follow for consistency between files and modules written by different teams. Automatic source code formatting tools may apply these conventions automatically.

Note: You can enforce these conventions automatically by running `terraform fmt`.

- Indent two spaces for each nesting level.
- When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs:

Question: 93 CertyIQ

Which of the following is not an action performed by terraform init?

A. Create a sample main.tf file
B. Initialize a configured backend
C. Retrieve the source code for all referenced modules
D. Load required provider plugins

Answer: A

Explanation:

Reference:
https://www.terraform.io/docs/cli/init/index.html

Question: 94 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

Reference:
https://www.packer.io/docs/templates/hcl_templates/functions
" target="_blank" style="word-break: break-all;">

The HCL language includes a number of built-in functions that you can call from within expressions to transform and combine values. The general syntax for function calls is a function name followed by comma-separated arguments in parentheses:

```
max(5, 12, 9)
```

Copy

For more details on syntax, see *Function Calls* on the Expressions page.

The HCL language does not support user-defined functions, and so only the functions built in to the language are available for use. The navigation for this section includes a list of all of the available built-in functions.

Question: 95 CertyIQ

How can you trigger a run in a Terraform Cloud workspace that is connected to a Version Control System (VCS) repository?

A. Only Terraform Cloud organization owners can set workspace variables on VCS connected workspaces

B. Commit a change to the VCS working directory and branch that the Terraform Cloud workspace is connected to

C. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces

D. Only Terraform Cloud organization owners can approve plans in VCS connected workspaces

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/cloud/vcs/index.html
" target="_blank" style="word-break: break-all;">

Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Although you can use many of Terraform Cloud's features without one, a VCS connection provides additional features and improved workflows. In particular:

- When workspaces are linked to a VCS repository, Terraform Cloud can automatically initiate Terraform runs when changes are committed to the specified branch.
- Terraform Cloud makes code review easier by automatically predicting how pull requests will affect infrastructure.
- Publishing new versions of a private Terraform module is as easy as pushing a tag to the module's repository.

We recommend configuring VCS access when first setting up an organization, and you might need to add additional VCS providers later depending on how your organization grows.

Configuring a new VCS provider requires permission to manage VCS settings for the organization. (More about permissions.)

Question: 96 CertyIQ

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

A.True
B.False

Answer: B

Explanation:

Reference:
https://www.terraform.io/docs/language/expressions/version-constraints.html

Question: 97 CertyIQ

Which statement describes a goal of infrastructure as code?

A.An abstraction from vendor specific APIs
B.Write once, run anywhere
C.A pipeline process to test and deliver software
D.The programmatic configuration of resources

Answer: D

Explanation:

Correct answer is D:The programmatic configuration of resources

Question: 98 CertyIQ

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

A.When a change is made to the resources via the Azure Cloud Console, the changes are recorded in a new state file

B.When a change is made to the resources via the Azure Cloud Console, Terraform will update the state file to reflect them during the next plan or apply

C.When a change is made to the resources via the Azure Cloud Console, the current state file will not be updated

D.When a change is made to the resources via the Azure Cloud Console, the changes are recorded in the current state file

Answer:

BC

Explanation:

B and C are both correct. Terraform plan DOES refresh the state file. extract from documentation . You don't need the (deprecated) refresh command to update state file. Newer option terraform plan -refresh lets you review changes to state file before modifying it. terraform refresh (deprecated) https://developer.hashicorp.com/terraform/cli/commands/refreshYou shouldn't typically need to use this command, because Terraform automatically performs the same refreshing actions as a part of creating a plan in both the terraform plan and terraform apply commands. This command is here primarily for backward compatibility, but we don't recommend using it because it provides no opportunity to review the effects of the operation before updating the state.

Question: 99 CertyIQ

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:

Reference:
https://github.com/hashicorp/terraform/issues/451

Question: 100 CertyIQ

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

A.Add node_count = var.node_count

B.Declare the variable in a terraform.tfvars file

C.Declare a node_count input variable for child module

D.Nothing, child modules inherit variables of parent module

Answer: C

Explanation:

C. Declare a node_ count input variable for child module. When passing variables from a parent module to a child module in Terraform, you need to explicitly declare the variables in both the parent and the child modules. In this case, you would need to declare an input variable for the child module that corresponds to the input variable declared in the parent module. So, the correct option is C. Option A is incorrect because it is using an example variable that is not related to the input variable mentioned in the question. Option B is incorrect because it refers to how to set the value of the variable, but not how to pass the value from the parent to the child module. Option D is also incorrect because child modules do not automatically inherit the variables of their parent modules.

Question: 101 CertyIQ

If a module declares a variable with a default, that variable must also be defined within the module.

A. True

B. False

Answer: B

Explanation:

If a module declares a variable with a default value, it is not necessary to define the variable within the module calling the module. The module will automatically use the default value if a value is not explicitly assigned to the variable in the calling module

Question: 102 CertyIQ

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

A. Environment Variables

B. Mark the variable as sensitive

C. A Terraform provider

D. A -var flag

Answer: C

Explanation:

We can mark the variable as sensitive and then pass the actual credentials using the --var tag.That would keep the secret outside the configuration file.

Reference:

https://www.linode.com/docs/guides/secrets-management-with-terraform/This article shows the answer

Question: 103 CertyIQ

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

A. sensitive

B. description

C. default

D. value

Answer: D

Explanation:

The only required argument when declaring a Terraform output is the value argument, which specifies the value of the output:

```

output "example"

value = "example output value"

```

The description argument is optional and can be used to provide additional information about the output:

```

output "example"

value = "example output value"

description = "An example output"

```

Question: 104 CertyIQ

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

A. With a Sentinel policy, which runs before every apply

B. By adding variables to each TFE workspace to ensure these settings are always enabled

C. With an S3 module with proper settings for buckets

D. Auditing cloud storage buckets with a vulnerability scanning tool

Answer: A

Explanation:

A. With a Sentinel policy, which runs before every apply.

Terraform Enterprise can enforce security controls through the use of Sentinel policies. Sentinel is a policy as code framework that integrates with Terraform Enterprise and can be used to enforce specific security controls. In this case, the Sentinel policy could check that all new S3 buckets are set to be private and encrypted at rest and prevent the Terraform apply from proceeding if the buckets do not meet this requirement. This ensures that the security control is automatically and proactively enforced every time Terraform makes changes to the infrastructure.

Question: 105 CertyIQ

Most Terraform providers interact with _____.

A. API

B. VCS Systems

C. Shell scripts

D. None of the above

Answer: A

Explanation:

Correct answer is A: API

Question: 106 CertyIQ

terraform validate validates that your infrastructure matches the Terraform state file.

A.True

B.False

Answer: B

Explanation:

Validate only checks the syntax

False B Validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus primarily useful for general verification of reusable modules, including correctness of attribute names and value types.It is safe to run this command automatically, for example as a post-save check in a text editor or as a test step for a re-usable module in a CI system

https://developer.hashicorp.com/terraform/cli/v1.1.x/commands/validate

Question: 107 CertyIQ

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

Explanation:

Import provisioned infrastructure to your state file

Question: 108 CertyIQ

FILL BLANK -
In the below configuration, how would you reference the module output vpc_id?

```
module "vpc" {
  source = "terraform-and-modules/vpc/aws"
  cidr = "10.0.0.0/16"
  name = "test-vpc"
}
```

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

Answer:

module. vpc. vpc_ id

Explanation:

Reference:

https://cloudcasts.io/course/terraform/community-vpc-module

Question: 109 CertyIQ

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: D

Explanation:

1. It's D.If you're using square brackets, then it should be like,ebs_block_device[*].volume_id
2. D. aws_instance.example.ebs_block_device.*.volume_idTo reference the Volume IDs associated with the ebs_block_device blocks in the given aws_instance resource, you can use the .* syntax to reference all elements of the list. The ebs_block_device blocks are represented as a list in the aws_instance resource, and each block has a volume_id attribute that you can reference.

Question: 110 CertyIQ

What does state locking accomplish?

A.Copies the state file from memory to disk

B.Encrypts any credentials stored within the state file

C.Blocks Terraform commands from modifying the state file

D.Prevents accidental deletion of the state file

Answer: C

Explanation:

If supported by your backend, Terraform will lock your state for all operations that could write state. This

prevents others from acquiring the lock and potentially corrupting your state.

Reference:

https://www.terraform.io/language/state/locking

Question: 111 CertyIQ

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:

Correct answer is B: Run terraform in it -upgrade

Question: 112 CertyIQ

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

A. True
B. False

Answer: B

Explanation:

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

Question: 113 CertyIQ

When you use a remote backend that needs authentication, HashiCorp recommends that you:

A. Use partial configuration to load the authentication credentials outside of the Terraform code
B. Push your Terraform configuration to an encrypted git repository
C. Write the authentication credentials in the Terraform configuration files
D. Keep the Terraform configuration files in a secret store

Answer: A

Explanation:

Reference:

https://www.terraform.io/language/settings/backends/configuration#credentials-and-sensitive-data

Question: 114 CertyIQ

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 terraform apply again immediately afterwards without changing any Terraform code?

A. Terraform will terminate and recreate the VM
B. Terraform will create another duplicate VM
C. Terraform will apply the VM to the state file
D. Nothing

Answer: C

Explanation:

Correct answer is C:Terraform will apply the VM to the state file.

Question: 115 CertyIQ

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
D. Stop and generate an error message about the missing instances

Answer: C

Explanation:

Correct answer is C:Rebuild only the instances that were deleted

Question: 116 CertyIQ

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

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

Answer: B

Explanation:

Terraform will prompt you to confirm that you want to destroy all the resources before proceeding. The prompt will ask you to enter "yes" or "no" to confirm or cancel the destruction of resources unless you add the "-auto-approve" flag to stop terraform from prompting you to confirm.

Question: 117 CertyIQ

Which type of block fetches or computes information for use elsewhere in a Terraform configuration?

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

Answer: D

Explanation:
1. it's D as you're using infrastructure already created - that's when you use "data". With "resource" you create new infrastructure.
2. Data sources allow data to be fetched or computed for use elsewhere in Terraform configuration. Use of data sources allows a Terraform configuration to build on information defined outside of Terraform, or defined by another separate Terraform configuration.

Question: 118 CertyIQ

You have just developed a new Terraform configuration for two virtual machines with a cloud provider. You would like to create the infrastructure for the first time.
Which Terraform command should you run first?

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

Answer: D

Explanation:
1. terraform init if you want to run it for the first time.
2. If it is first time then Terraform init command.

Question: 119 CertyIQ

All modules published on the official Terraform Module Registry have been verified by HashiCorp.

A.True
B.False

Answer: B

Explanation:

Answer is False. 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/verifiedhttps://www.terraform.io/registry/modules/publish

Question: 120 CertyIQ

You have to initialize a Terraform backend before it can be configured.

A.True
B.False

Answer: B

Explanation:

B. False The backend configuration is specified within the Terraform configuration files, typically in the main.tf file. You don't have to initialize a backend before configuring it; instead, you provide the backend configuration within the Terraform files, and then run terraform init. The initialization process will set up the backend as per the configuration provided. So, the backend configuration is done first, and then you initialize Terraform, which sets up the backend accordingly.

Question: 121 CertyIQ

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

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

Answer: A

Explanation:

Correct answer is A:Cloud infrastructure

Question: 122 CertyIQ

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

Explanation:

When you execute commands like terraform apply, terraform taint, or terraform destroy, Terraform needs to access and update the state file to perform the desired actions. The backend determines how Terraform interacts with the state file during these operations.

Question: 123 CertyIQ

What does Terraform use .terraform.lock.hcl file for?

A. Tracking provider dependencies

B. There is no such file

C. Preventing Terraform runs from occurring

D. Storing references to workspaces which are locked

Answer: A

Explanation:

A. Tracking provider dependencies

The .terraform. lock. hcl file is used to track provider dependencies and their exact versions. This file is automatically generated by Terraform when you run terraform init. It locks the versions of the providers used in your configuration, ensuring that subsequent Terraform runs use the same provider versions for consistency and reproducibility across environments. This file should be committed to your version control system to maintain consistency across team members and environments.

Question: 124 CertyIQ

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

A. Use the terraform state rm command to remove the VM from state file

B. Use the terraform taint command targeting the VMs then run terraform plan and terraform apply

C. Use the terraform apply command targeting the VM resources only

D. Delete the Terraform VM resources from your Terraform code then run terraform plan and terraform apply

Answer: B

Explanation:

B. for v0.15.2 and later, you can also use "terraform apply -replace=ADDRESS" to achieve the same result.

Question: 125 CertyIQ

How do you specify a module's version when publishing it to the public Terraform Module Registry?

A. The module's configuration page on the Terraform Module Registry
B. Terraform Module Registry does not support versioning modules
C. The release tags in the associated repo
D. The module's Terraform code

Answer: C

Explanation:

C. The release tags in the associated repo. When publishing a module to the Terraform Module Registry, the module version is typically specified using Git tags in the associated Git repository. Terraform retrieves the module from the registry using the provider's source argument, which includes the module's owner, name, and version. By specifying the version in the source argument of your module, Terraform will download that specific version of the module from the registry.

Question: 126 CertyIQ

Terraform plan updates your state file.

A.True
B.False

Answer: B

Explanation:
1. B. FalseThe terraform plan command does not update your state file. Instead, it generates an execution plan by comparing the desired state (defined in your Terraform configuration files) with the actual state (stored in the state file) of your infrastructure. The plan shows the changes that Terraform will make to the infrastructure without actually applying those changes. The state file is only updated when you run terraform apply, which applies the changes to your infrastructure and updates the state file to reflect the new state.
2. The terraform plan command does not change the state of your infrastructure. It creates an execution plan that lets you preview the changes that Terraform plans to make to your infrastructure

Question: 127 CertyIQ

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:
1. -check -recursive should check all formatting without modifying the configuration files.
2. terraform fmt takes followong optionsOptions: -list=falseDon't list files whose formatting differs (always

disabled if using STDIN) -write=false Don't write to source files (always disabled if using STDIN or -check) -diff Display diffs of formatting changes -check Check if the input is formatted. Exit status will be 0 if all input is properly formatted and non-zero otherwise. -no-color If specified, output won't contain any color. -recursive Also process files in subdirectories. By default, only the given directory (or current directory) is processed.

Question: 128 CertyIQ

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

A.null-exıμ

B.local-exec

C.remote-exec

D.file

Answer: C

Explanation:

remote-exec to run anything on the deployed virtual machine. local exec is to run anything on the system where terraform is running

The remote-exec provisioner invokes a script on a remote resource after it is created. To invoke a local process, see the local-exec provisioner instead.

Reference:

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

Question: 129 CertyIQ

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

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

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

```
Error: Reference to undeclared output value

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

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

A.Define the attribute vnet_id as a variable in the networking module

B.Change the referenced value to module.my_network.outputs.vnet_id

C.Define the attribute vnet_id as an output in the networking module

D.Change the referenced value to my_network.outputs.vnet_id

Answer: C

Explanation:

1. In a parent module, outputs of child modules are available in expressions as module.<MODULE NAME>.<OUTPUT NAME>. For example, if a child module named web_server declared an output named instance_ip_addr, you could access that value as module.web_server.instance_ip_addr.Answer is C
2. All looks good except that the output in the child module is missing.

Question: 130 CertyIQ

You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name the instance resource "main".
Which of these is the correct way to define the output value using HCL2?
A.

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

B.

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

Answer: A

Explanation:

correct answer is A

Question: 131 CertyIQ

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: C

Explanation:

As additional resources are required, more tickets are submitted

Question: 132 CertyIQ

Which of the following statements about Terraform modules is not true?

A. Modules must be publicly accessible
B. Modules can be called multiple times
C. Module is a container for one or more resources
D. Modules can call other modules

Answer: A

Explanation:

A: In addition to modules from the local filesystem, Terraform can load modules from a public or private registry. Also, members of your organization might produce modules specifically crafted for your own infrastructure needs. Source: https://www.terraform.io/language/modules

Question: 133 CertyIQ

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

A. tuple
B. set
C. ma€
D. list

Answer: C

Explanation:

Correct answer is C: ma€

Question: 134 CertyIQ

You have used Terraform to create an ephemeral development environment in the cloud and are now ready to destroy all the infrastructure described by your
Terraform configuration. To be safe, you would like to first see all the infrastructure that will be deleted by Terraform.
Which command should you use to show all of the resources that will be deleted? (Choose two.)

A. Run terraform plan -destroy
B. Run terraform show -destroy
C. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval
D. Run terraform show -destroy

Answer: AC

Explanation:

A. Run terraform plan -destroy

C. Run terraform destroy and it will first output all the resources that will be deleted before prompting for approval

Question: 135 CertyIQ

When do you need to explicitly execute terraform refresh?

A. Before every terraform plan
B. Before every terraform apply
C. Before every terraform import
D. None of the above

Answer: D

Explanation:

D. None of the above.

Refresh is used to check backward compatibility to read the infrastructure objects, and update the state file. Terraform performs refresh by itself during every plan and apply commands. So, it is recommended that we not perform refresh often.

Question: 136 CertyIQ

All Terraform Cloud tiers support team management and governance.

A.True
B.False

Answer: B

Explanation:
1. 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. Source: https://www.terraform.io/cloud-docs/overview
2. "Cloud Free" does not provide "Team management". In below link, click "Unified workflow management" under "Features" and search the whole page (Ctrl-F) with "Team management".
https://www.hashicorp.com/products/terraform/pricing

Question: 137 CertyIQ

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

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

Answer: C

Explanation:

1. This should be it : "The ability to reuse best practice configurations and settings"
2. C. The ability to reuse best practice configurations and settings.An operations team that uses infrastructure as code has the advantage of being able to reuse best practice configurations and settings across their infrastructure. This enables the team to quickly and easily provision new resources, and ensures that resources are configured consistently across the infrastructure. By defining infrastructure as code, the team can also automate the provisioning and configuration of resources, which helps reduce the risk of manual errors and frees up time for more important tasks.

Question: 138 CertyIQ

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

B.terraform apply

C.terraform refresh

Answer: A

Explanation:

1. I guess it had to be terraform state mv .....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.
2. A is correct but the command is wrong it should beterraform state mv <old_name> <new_name>

Question: 140 CertyIQ

Terraform variable names are saved in the state file.

A.True
B.False

Answer: B

Explanation:

B. False. Terraform variable names are not saved in the state file. The state file contains information about the resources that Terraform manages, such as their current state and metadata. Variables are used in your Terraform configuration to parameterize your code and make it more reusable, but they are not saved in the state file. Instead, variable values are provided at runtime, either by passing them on the command line, through environment variables, or by using default values defined in your configuration.

Question: 141 CertyIQ

Terraform Cloud is available only as a paid offering from HashiCorp.

A. True
B. False

Answer: B

Explanation:

"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."

Question: 142 CertyIQ

Which of the following is not a way to trigger terraform destroy?

A. Using the destroy command with auto-approve
B. Running terraform destroy from the correct directory and then typing "yes" when prompted in the CLI
C. Passing --destroy at the end of a plan request
D. Delete the state file and run terraform apply

Answer: D

Explanation:

Correct answer is D:Delete the state file and run terraform apply

Question: 143 CertyIQ

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:

B is correct since terraform is used to deploy the infrastructure, not to troubleshoot it.

Question: 144 CertyIQ

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: A

Explanation:

fileset("id_rsa.pub")

Question: 145 CertyIQ

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

A. Provision resources for on-premises infrastructure services

B. Simplify API interactions

C. Provision resources for public cloud infrastructure services

D. Enforce security and compliance policies

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

Answer: ABC

Explanation:

A. Provision resources for on-premises infrastructure services

B. Simplify API interactions

C. Provision resources for public cloud infrastructure services

Question: 146 CertyIQ

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

A.True

B.False

Answer: B

Explanation:

You cannot reference a resource created with for_each using a splat (*) expression. Resources that use the for_each argument will appear in expressions as a map of objects, so you can't use splat expressions and you have to use a for expression loop.For example, the following code will create two EC2 instances:resource "aws_instance" "web" for_each = [1, 2] ami = "ami-0123456789abcdef0" instance_type = "t2.micro" You cannot reference the EC2 instances created by this code using a splat expression, such as *aws_instance.web. Instead, you would have to use a for expression loop, such as:for i, instance in aws_instance.web.items() # Do something with the instance

Question: 147 CertyIQ

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:

When Terraform evaluates a configuration, it automatically creates a graph of all the resources and their dependencies. Terraform uses this graph to determine the order in which resources need to be created, updated, or destroyed. Terraform looks at resource dependencies based on implicit and explicit relationships between resources, which are defined using properties like count, for_ each, depends_ on, and others. Terraform can also use the state file to track dependencies between resources and maintain the order of operations during updates.

Reference:

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

Question: 148 CertyIQ

Which parameters does terraform import require? (Choose two.)

A.Path

B.Provider

C.Resource ID

D.Resource address

Answer: CD

Explanation:

The command is terraform import <resource address> <resource id>

Reference:

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

Question: 149 CertyIQ

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

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

Answer: D

Explanation:

D. terraform init. When configuring a new Terraform backend, the state file needs to be migrated to the new backend so that it can be used to manage infrastructure state going forward. The terraform in it command is used to initialize a new backend and migrate the state file to that backend.

Question: 150 CertyIQ

What does this code do?

```
terraform {
  required_providers {
    aws = "~> 3.0"
  }
}
```

A.Requires any version of the AWS provider >= 3.0 and < 4.0
B.Requires any version of the AWS provider >= 3.0
C.Requires any version of the AWS provider after the 3.0 major release, like 4.1
D.Requires any version of the AWS provider > 3.0

Answer: A

Explanation:

A. Requires any version of the AWS provider >= 3.0 and < 4.0.In Terraform, provider version constraints can be specified using version ranges. The version constraint ~> 3.0 specifies that the configuration requires any version of the AWS provider that is greater than or equal to version 3.0, but less than version 4.0.

Question: 151 CertyIQ

What does terraform refresh modify?

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

Answer: B

Explanation:

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

Reference:

https://www.terraform.io/cli/commands/refresh

Question: 152 CertyIQ

Which of the following is not valid source path for specifying a module?

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

Answer: A

Explanation:

Correct answer is A: source = "./modulel version=v1.0.0"

Question: 153 CertyIQ

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:

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

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

The terraform apply command in Terraform is used to create or modify infrastructure resources according to the Terraform configuration in the current working directory or workspace.

Question: 154 CertyIQ

Which of the following statements about local modules is incorrect?

A. Local modules are not cached by terraform init command
B. Local modules are sourced from a directory on disk
C. Local modules support versions
D. All of the above (all statements above are incorrect)
E. None of the above (all statements above are correct)

Answer: C

Explanation:

Answer is C.

In particular, modules sourced from local file paths do not support version; since they're loaded from the same source repository, they always share the same version as their caller.

Reference:

https://www.terraform.io/language/modules/syntax

Question: 155 CertyIQ

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

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

Answer: CE

Explanation:

C. You can version your infrastructure configuration.

E. It allows you to automate infrastructure provisioning.

Terraform's implementation of infrastructure as code has the following characteristics:

Option C is true - you can version your infrastructure configuration using version control systems like Git. Terraform supports multiple version control backends such as Git, Subversion, and Mercurial, and allows

users to manage multiple versions of infrastructure code in the same repository.

Option E is true - Terraform allows you to automate infrastructure provisioning. Terraform uses configuration files to describe the desired state of infrastructure, and then automatically provisions and configures the infrastructure to match that state.

Question: 156 CertyIQ

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

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

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

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

Answer: D

Explanation:

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

Reference:

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

Question: 157 CertyIQ

Which of the following is the safest way to inject sensitive values into a Terraform Cloud workspace?

A.Write the value to a file and specify the file with the -var-file flag
B.Set a value for the variable in the UI and check the "Sensitive" check box

C.Edit the state file directly just before running terraform apply

D.Set the variable value on the command line with the -var flag

Answer: B

Explanation:

B. Set a value for the variable in the UI and check the "Sensitive" check box. When working with Terraform Cloud workspaces, the safest way to inject sensitive values into a Terraform Cloud workspace is to set a value for the variable in the UI and check the "Sensitive" check box. This will ensure that the value is stored securely and not visible in plain text in the Terraform Cloud UI or API. Option A, writing the value to a file and specifying the file with the -var-file flag, may be less secure because the file could potentially be accessed by unauthorized users. Option C, editing the state file directly just before running terraform apply, is not a best practice and could result in data loss or corruption. Option D, setting the variable value on the command line with the -var flag, could result in the sensitive value being stored in plain text in the command history or other logs, which could be accessed by unauthorized users.

Question: 158 CertyIQ

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

A.True

B.False

Answer: B

Explanation:

Terraform plan is not mandatory. It just lays our the execution plan. It is an optional command. So, the answer is B. False

Question: 159 CertyIQ

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

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

A.resource.kubernetes_namespace.example.name

B.kubernetes_namespace.test.name

C.kubernetes_namespace.example.name

D.data.kubernetes_namespace.name

E.None of the above

Answer: C

Explanation:

Reference:

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

Question: 160 CertyIQ

A Terraform output that sets the "sensitive" argument to true will not store that value in the state file.

A.True
B.False

Answer: B

Explanation:

B. False. A Terraform output that sets the "sensitive" argument to true will still store that value in the state file. The "sensitive" argument is used to prevent the value from being displayed in plain text in the Terraform CLI output, Terraform Cloud UI, and other locations where output values may be displayed. However, the value will still be stored in the Terraform state file, which is used to track the current state of the infrastructure.It is important to be aware that while the "sensitive" argument can help to prevent accidental exposure of sensitive values, it is not a substitute for proper security practices such as role-based access control and data encryption.

Question: 161 CertyIQ

Which are forbidden actions when the Terraform state file is locked? (Choose three.)

A. terraform destroy
B. terraform fmt
C. terraform state list
D. terraform apply
E. terraform plan
F. terraform validate

Answer: ADE

Explanation:

A. terraform destroy

D. terraform apply

E. terraform plan

When the Terraform state file is locked, there are certain actions that are not allowed as they can result in data corruption or loss. The actions that are forbidden when the state file is locked include:

A. terraform destroy: This command destroys all resources defined in the configuration file, which could cause data loss if the state file is locked.

D. terraform apply: This command applies changes to the infrastructure as defined in the configuration file, which could cause data corruption if the state file is locked.

E. terraform plan: This command creates an execution plan of what changes Terraform will apply to the infrastructure, which could cause data corruption if the state file is locked.

Actions such as terraform fmt, terraform state list, and terraform validate are read-only operations that do not modify the state file, and can be performed safely even when the state file is locked.

Question: 162 CertyIQ

Terraform installs its providers during which phase?

A. Plan
B. Init
C. Refresh
D. All of the above

Answer: B

Explanation:

Terraform installs providers at init phase

Question: 163 CertyIQ

When does Sentinel enforce policy logic during a Terraform Enterprise run?

A. Before the plan phase
B. During the plan phase
C. Before the apply phase
D. After the apply phase

Answer: C

Explanation:

Before the apply phase

Question: 164 CertyIQ

What is the purpose of a Terraform workspace in either open source or enterprise?

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

Answer: A

Explanation:

A. Workspaces allow you to manage collections of infrastructure in state files.

The purpose of a Terraform workspace, both in open source and enterprise, is to allow users to manage multiple "instances" of infrastructure within the same configuration codebase. Each workspace is a separate instance of a Terraform state file that can be managed independently. This means that a single Terraform configuration can be used to manage multiple sets of resources, with each set of resources represented by a separate workspace.

Workspaces are useful when there is a need to manage multiple versions of the same infrastructure in the same configuration codebase, such as different environments (dev, staging, prod) or different regions. By creating a separate workspace for each instance of infrastructure, users can manage them independently without causing conflicts or overwriting state.

Option B, C, and D are not correct. While workspaces may be used in conjunction with logical separation of business units, grouping multiple infrastructure security policies, or providing limited access to a cloud environment, they are not the primary purpose of workspaces in Terraform.

Question: 165 CertyIQ

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

A. Append ?ref=v1. 0. 0 argument to the source path
B. Add version = "1.0.0" parameter to module block
C. Nothing " modules stored on GitHub always default to version 1.0.0
D. Modules stored on GitHub do not support versioning

Answer: A

Explanation:

If the source is from terrform registry then we can use version = "1.0.0"

when using git as the source we have to use ?ref=1.0.0 the exact git path give in in the question is given in this terraform official documentation .

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

Question: 166 CertyIQ

Changing the Terraform backend from the default "local" backend to a different one after doing your first terraform apply is:

A.Mandatory
B.Optional
C.Impossible
D.Discouraged

Answer: B

Explanation:

B. Optional , the mandatory using terraform init

Question: 167 CertyIQ

You have 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

Answer: B

Explanation:

Your plan has been changed. So its best to run the plan again to make sure the desired state is achieved.

Question: 168 CertyIQ

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.

Question: 169 CertyIQ

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.Clone the repository containing your infrastructure code and then run the code

B.Use the public cloud console to make the change after a database record has been approved

C.Make the change programmatically via the public cloud CLI

D.Make the change via the public cloud API endpoint

E.Submit a pull request and wait for an approved merge of the proposed changes

Answer: E

Explanation:

E. Submit a pull request and wait for an approved merge of the proposed changes. As a member of an operations team that uses infrastructure as code (I a C) practices, the best practice for making a change to an infrastructure stack running in a public cloud is to submit a pull request (PR) and wait for an approved merge of the proposed changes. This approach follows the best practice of using version control for infrastructure code and collaborating through pull requests to review and approve changes. By following this process, changes to the infrastructure stack are tracked, reviewed, and approved in a controlled and auditable way.

Question: 170 CertyIQ

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 Graph Viz to generate charts.

Question: 171 CertyIQ

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

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

Answer: A

Explanation:

Correct answer is A: Using environment variables

Question: 172 CertyIQ

Running terraform fmt without any flags in a directory with Terraform configuration files will check the formatting of those files without changing their contents.

A. True
B. False

Answer: B

Explanation:

The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style.

Question: 173 CertyIQ

terraform init retrieves the source code for all referenced modules.

A. True
B. False

Answer: A

Explanation:

A. True. Terraform in it retrieves the source code for all referenced modules. When Terraform in it is run, it will automatically download the source code for any modules referenced in the configuration files. This ensures that the correct version of the module is available for use when Terraform applies changes to the infrastructure. The source code is then cached locally so that it can be used in future runs without having to re-download it. It's important to notice that this command also initializes the backend and sets the backend configuration if specified.

Question: 174 CertyIQ

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:

Correct answer is B: Terraform will destroy the virtual machine

Question: 175 CertyIQ

Which configuration consistency errors does terraform validate report?

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

Answer: D

Explanation:

D. Declaring a resource identifier more than once.

Terraform validate is a command that checks the syntax and structure of the Terraform configuration files. It reports any configuration consistency errors it finds before attempting to create or modify any resources. The command checks for a variety of errors, including syntax errors, type errors, and missing required arguments. However, it does not check for differences between local and remote state, or whether a Terraform module is the latest version.

The only option that describes an error that Terraform validate checks for is D. Declaring a resource identifier more than once. This error occurs when the same resource identifier is declared multiple times in the same configuration file, which can lead to conflicts and unexpected behaviour. Terraform validate checks for this error and reports it if it is found in the configuration files.

Question: 176 CertyIQ

In Terraform HCL, an object type of object(name=string, age=number) would match this value:
A.

```
{
  name = "John"
  age = fifty two
}
```

B.

```
{
  name = "John"
  age = 52
}
```

Answer: B

Explanation:

Age = 52 Only number.

Question: 177 CertyIQ

Where can Terraform not load a provider from?

A.Source code
B.Plugins directory

C.Official HashiCorp distribution on releases.hashicorp.com

D.Provider plugin cache

Answer: A

Explanation:

A- no info about TF loading a provider directly from source code, so this must be the correct answer. B- In the plugin directory are stored the providers, so TF could load providers from it C- releases.hashicorp.com is a valid source where providers can be downloaded from D- If the plugin provider cache is enabled eventually a provider could be loaded from it.

Question: 178 CertyIQ

Which of the following locations can Terraform use as a private source for modules? (Choose two.)

A.Internally hosted SCM (Source Control Manager) platform

B.Public Terraform Module Registry

C.Private repository on GitHub

D.Public repository on GitHub

Answer: AC

Explanation:

A. Internally hosted SCM (Source Control Manager) platform C. Private repository on GitHub Terraform can use the following locations as a private source for modules: A. Internally hosted SCM (Source Control Manager) platform: Terraform allows you to specify a private SCM (Source Control Manager) platform like GitLab, Bitbucket, etc as a source for modules. This allows you to host your own modules within your organization's network, and use them privately. C. Private repository on GitHub: Terraform allows you to specify a private repository on GitHub as a source for modules. This allows you to host your own modules on GitHub and use them privately.

Question: 179 CertyIQ

Why should secrets not be hard coded into Terraform code? (Choose two.)

A.It makes the code less reusable.

B.Terraform code is typically stored in version control, as well as copied to the systems from which it's run. Any of those may not have robust security mechanisms.

C.The Terraform code is copied to the target resources to be applied locally and could expose secrets if a target resource is compromised.

D.All passwords should be rotated on a quarterly basis.

Answer: AB

Explanation:

1. AB, I don't think a terraform code is copied to any place(local, backend, any modules etc.,) But, The values of the variables are rendered into the state file. where the key/secrets are exposed

2. A and B

Question: 180 CertyIQ

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:

Answer C: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.

Question: 181 CertyIQ

When should Terraform configuration files be written when running terraform import on existing infrastructure?

A. Infrastructure can be imported without corresponding Terraform code

B. Terraform will generate the corresponding configuration files for you

C. You should write Terraform configuration files after the next terraform import is executed

D. Terraform configuration should be written before terraform import is executed

Answer: D

Explanation:

1. The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration.Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.Source: https://www.terraform.io/cli/import

2. D. Terraform configuration should be written before terraform import is executedWhen running the terraform import command on existing infrastructure, Terraform uses the configuration files to know how to interact with the resources. Without the configuration files, Terraform does not know how to interact with the resources and what to import.Therefore, Terraform configuration files should be written before terraform import is executed.A. Infrastructure can be imported without corresponding Terraform code is not true.B. Terraform will generate the corresponding configuration files for you is not true.C. You should write Terraform configuration files after the next terraform import is executed is not true.It's important to note that Terraform import is not a replacement for creating Terraform configuration files, but it helps to discover and populate the state with existing resources.

Question: 182 CertyIQ

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:

Reference:
https://www.terraform.io/language/functions

Question: 183 CertyIQ

Why does this backend configuration not follow best practices?

```
terraform {
  backend "s3" {
    bucket     = "terraform-state-prod"
    key        = "network/terraform.tfstate"
    region     = "us-east-1"
    access_key = "AKIAIOSFODNN7EXAMPLE"
    secret_key = "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
  }

  required_providers {
       aws = {
         source  = "hashicorp/aws"
         version = "~> 3.38"
       }
     }

     required_version = ">= 0.15"
}
```

A. You should not store credentials in Terraform Configuration

B. You should use the local enhanced storage backend whenever possible

C. An alias meta-argument should be included in backend blocks whenever possible

D. The backend configuration should contain multiple credentials so that more than one user can execute terraform plan and terraform apply

Answer: A

Explanation:

You should not store credentials in Terraform Configuration

Question: 184 CertyIQ

Open source Terraform can only import publicly-accessible and open-source modules.

A. True

B. False

Answer: B

Explanation:

1. 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
2. B. False.Open source Terraform can import both publicly-accessible and private modules from various sources, including Terraform Registry, GitHub, GitLab, Bitbucket, and others. However, the support for private modules is limited in the open source version, while it's fully supported in Terraform Enterprise.

Question: 185 CertyIQ

What does terraform destroy do?

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

Answer: A

Explanation:

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

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

Question: 186 CertyIQ

While attempting to deploy resources into your cloud provider using Terraform, you begin to see some odd behavior and experience sluggish responses. In order to troubleshoot you decide to turn on Terraform debugging. Which environment variables must be configured to make Terraform's logging more verbose?

A.TF_LOG_LEVEL
B.TF_LOG_FILE
C.TF_LOG
D.TP_LOG_PATH

Answer: C

Explanation:

Reference:
https://www.terraform.io/internals/debugging

Question: 187 CertyIQ

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

A.The team is asked to build a reusable code base that can deploy resources into any AWS region

B.The team is asked to manage a new application stack built on AWS-native services

C.The organization decides to expand into Azure and wishes to deploy new infrastructure using their existing codebase

D.The DevOps team is tasked with automating a manual provisioning process

Answer: C

Explanation:

C. The organization decides to expand into Azure and wishes to deploy new infrastructure using their existing code base AWS Cloud Formation is a service specifically designed for provisioning AWS resources. If the organization decides to expand into Azure and deploy new infrastructure using the existing codebase, the DevOps team will face a challenge, as Cloud Formation does not support Azure or other cloud providers.

https://developer.hashicorp.com/terraform/intro/vs/cloudformation

Question: 188 CertyIQ

You cannot install third party plugins using terraform init.

A.True

B.False

Answer: B

Explanation:

B. False Terraform does support installing third-party plugins, but the process is not as straightforward as it is for official providers. Starting with Terraform 0.13, you can install third-party providers using the terraform in it command if you have properly configured the required_ provider block and set up the provider source.

Question: 189 CertyIQ

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

A.Save a generated execution plan to apply later

B.Execute a plan in a different workspace

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

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

Answer: AC

Explanation:

Reference:
https://learn.hashicorp.com/tutorials/terraform/plan

Question: 190 CertyIQ

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

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

Answer: CD

Explanation:

C. Versioned configuration files D. Docker files Examples of infrastructure as code are: C. Versioned configuration files: Infrastructure as code is when infrastructure is defined and managed using code, and configuration files that are written in a programming language, such as JSON or YAML, and stored in a version control system like Git, allowing for versioning and collaboration. D. Docker files: Docker files are a type of infrastructure as code that describe how to build a container image. They are written in a simple programming language and can be versioned and stored in a version control system.

Question: 191 CertyIQ

FILL BLANK -
You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

Answer: terraform init

Question: 192 CertyIQ

When using a module from the public Terraform Module Registry, the following parameters are required attributes in the module block. (Choose two.)

A. Each of the module's required inputs
B. The module's source address
C. Terraform Module Registry account token
D. Each of the module's dependencies (example: submodules)
E. The version of the module

Answer: AB

Explanation:
1. The answer is A,B .. version isnt mendatory: " " "Registry modules support versioning. You can provide a specific version as shown in the above examples, or use flexible version constraints." " "
2. Version is not mandatory. Only mandatory is "source" and the input vars that are mandatory for the module to operate.

Question: 193 CertyIQ

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 init -upgrade

B. terraform apply -upgrade

C. terraform refresh -upgrade

D. terraform providers -upgrade

Answer: A

Explanation:

terraform init -upgrade

Question: 194 CertyIQ

You can access state stored with the local backend by using the terraform_remote_state data source.

A. True

B. False

Answer: A

Explanation:

The terraform_ remote_ state data source allows you to retrieve outputs from the state of another Terraform configuration, which can be stored in a local or remote backend. To use the terraform_ remote_ state data source with a local backend, you would define the path to the state file in the backend configuration block of your Terraform configuration, and then use that path in the data "terraform_ remote_ state" block to retrieve the desired output values.

Question: 195 CertyIQ

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

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

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

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

A. Terraform will fail with an error because the state file is no longer accurate.

B. Terraform will change the load balancer port to 80, and then change it back to 443.

C. Terraform will not make any changes to the Load Balancer and will update the state file to reflect any changes made.

D. Terraform will change the port back to 80 in your code.

Answer: C

Explanation:

As the state is refreshed during the "apply" no changes will be made on the cloud. Terraform will rather

update it state file.

Question: 196 CertyIQ

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

A.True

B.False

Answer: A

Explanation:

It is true because Terraform Cloud can monitor the linked version control repository for changes and automatically trigger a speculative plan run in response to each commit or merge. This allows users to quickly see the expected changes that would result from the proposed change, without actually applying those changes. Speculative plan runs can be a useful tool for catching errors early in the development process and avoiding potentially costly mistakes.

Reference:

https://developer.hashicorp.com/terraform/cloud-docs/run/ui#automatically-starting-runs

Question: 197 CertyIQ

You have some Terraform code and a variable definitions file named dev.auto.tfvars that you tested successfully in the dev environment. You want to deploy the same code in the staging environment with a separate variable definition file and a separate state file.

Which two actions should you perform? (Choose two.)

A.Copy the existing terraform.tfstate file and save it as staging.terraform.tfstate

B.Write a new staging.auto.tfvars variable definition file and run Terraform with the var-file=" staging.auto.tfvars" flag

C.Create a new Terraform workspace for staging

D.Create a new Terraform provider for staging

E.Add new Terraform code (*.tf files) for staging in the same directory

Answer: BC

Explanation:

B and C are correct answers. same tf state file can be used in multiple workspaces to provision infrastructure and each workspace has its own tf var file matching workspace prefix.

Question: 198 CertyIQ

The _____ determines how Terraform creates, updates, or deletes resources.

A.Terraform configuration

B.Terraform core

C.Terraform provider

D.Terraform provisioner

Answer: C

Explanation:

1. The question specifically states "how". The provider is the only component of Terraform that know HOW to create, update, delete a resource, as it knows all the specifics.
2. The Terraform provider determines how Terraform creates, updates, or deletes resources.Providers are responsible for translating Terraform configurations into API requests that manipulate resources in the target environment. Each provider implements the necessary operations for a specific type of infrastructure, such as AWS, Azure, or Google Cloud Platform.When Terraform applies a configuration, it reads the provider information from the configuration and uses it to interact with the target environment. The provider then manages the lifecycle of the resources by creating, updating, or deleting them as needed.Note that while the Terraform configuration defines the desired state of the resources, and the Terraform core is responsible for managing the planning and execution of changes, it is ultimately the provider that determines how those changes are implemented in the target environment.

Question: 199 CertyIQ

Terraform destroy is the only way to remove infrastructure.

A.True

B.False

Answer: B

Explanation:

B. by commenting the resource block in the configuration file will also destroy the resources

Question: 200 CertyIQ

Which of the following is the correct way to pass the value in the variable num_servers into a module with the input servers in HCL2?

A.servers - var.num_servers

B.servers - num_servers

C.servers - var(num_servers)

D.$(var.num_servers)

Answer: A

Explanation:

module "example" source = "path/to/module" servers = var. num_ servers

The correct way to pass the value in the variable num_ servers into a module with the input servers in HCL2 is as follows: module "example" source = "./modules/example" servers = var. num_ servers

Question: 201 CertyIQ

Which of the fallowing commands would you use to access all of the attributes and details of a resource managed by Terraform?

A.terraform state list

B.terraform state show

C.terraform get

D.terraform state list

Answer: B

Explanation:

The terraform state list command shows the resource addresses for every resource Terraform knows about in a configuration, optionally filtered by partial resource address. The terraform state show command displays detailed state data about one resource.

Question: 202 CertyIQ

How would you be able to reference an attribute from the vsphere_datacenter data source for use with the datacenter_id argument within the vsphere_folder resource in the following configuration?

```
data "vsphere_datacenter" "dc" {}

resource "vsphere_folder" "parent" {
            path = "Production"
            type = "vm"
  datacenter_id = _____________________
}
```

A.data.dc.id

B.data.vsphere_datacenter.dc

C.vsphere_datacenter.dc.id

D.data.vsphere_datacenter.dc.id

Answer: D

Explanation:

D:data "azure rm_ resource_ group" "example" name = "existing" output "id" value = data.azurerm_resource_group.example.id

Question: 203 CertyIQ

You decide to move a Terraform state file to Amazon S3 from another location. You write the code below into a file called backend.tf.

```
terraform {
  backend "s3" {
    bucket - "my-tf-bucket"
    region = "us-east-1"
  }
}
```

Which command will migrate your current state file to the new S3 remote backend?

A.terraform state
B.terraform init
C.terraform refresh
D.terraform push

Answer: B

Explanation:

terraform in it migrates the state.

Question: 204 CertyIQ

You want to tag multiple resources with a string that is a combination of a generated random_id and a variable.

How should you use the same value in all these resources without repeating the random_id and variable in each resource?

A.Local values
B.Data source
C.Modules
D.Outputs

Answer: A

Explanation:

A is correct: Local values: A local value assigns a name to an expression, so you can use the name multiple times within a module instead of repeating the expression.

Reference:

https://developer.hashicorp.com/terraform/language/values/locals

Question: 205 CertyIQ

Which of the following is not a benefit of adopting infrastructure as code?

A.Interpolation
B.Reusability of code
C.Versioning

D.Automation

Answer: A

Explanation:

Correct answer is A:Interpolation

Question: 206 CertyIQ

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

A.True
B.False

Answer: B

Explanation:

Specifying a versions is not mandatory. When a version is not specified, Terraform just downloads the latest version.

Question: 207 CertyIQ

While deploying a virtual machine, the first launch user_data script fails due to race condition with another resource deployed during the same Terraform run.

What is the least disruptive method to correct the issue?

A.Run terraform taint against the virtual machine's resource name, then terraform apply
B.Restart the virtual machine from the cloud portal
C.Run terraform apply again
D.Run terraform destroy then terraform apply

Answer: A

Explanation:

The script deployment will not have been tracked within Terraform - and will therefore not re-run on the next 'apply' command. The resource the script is running on will have to be redeployed to trigger the script to run a second time - the least disruptive way do achieve this is with the 'taint' command.

Question: 208 CertyIQ

The public Module Registry is free to use.

A.True
B.False

Answer: A

Explanation:

Both public and private registries are free. "Terraform Cloud includes a private registry that is available to all accounts, including free organizations."https://developer.hashicorp.com/terraform/registry/private#terraform-cloud-private-registry"Anyone can publish and consume providers, modules, and policies on the public Terraform Registry

Reference:

"https://developer.hashicorp.com/terraform/registry

Question: 209 CertyIQ

Both Terraform Cloud and Terraform Enterprise support policy as code (Sentinel).

A.True
B.False

Answer: A

Explanation:

True is a right answer.

Question: 210 CertyIQ

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.Collection functions
C.Dynamic blocks
D.Count arguments

Answer: C

Explanation:

The correct answer is C. Dynamic Blocks. Dynamic blocks allow you to define multiple data disks as nested blocks inside the resource block for a virtual machine by using the values in a variable. Dynamic blocks are used to create multiple resource instances from a single configuration block by iterating over the values of a list or map.

Question: 211 CertyIQ

Which of the following module source paths does not specify a remote module?

A. source = “./modules/consul”
B. source = “:hashicorp/example.git”

C. source = “github.com/hashicorp/example”

D. source = “hashicorp/consul/aws”

Answer: C

Explanation:

source = “github.com/hashi corp/example”

Question: 212 CertyIQ

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 function could you use to select the largest number from the list?

A. max(numcpus)

B. ceil(numcpus)

C. top(numcpus)

D. high[numcpus]

Answer: A

Explanation:

"max takes one or more numbers and returns the greatest number from the set."

Question: 213 CertyIQ

Variables declared within a module are accessible outside of the module.

A. True

B. False

Answer: B

Explanation:

B. False.

Variables declared within a module have module-level scope and can only be accessed within the module. If a variable in a module needs to be accessed outside of the module, it must be exposed as an output.

Question: 214 CertyIQ

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

A. list

B. map

C. array

D. string

Answer: C

Explanation:

Array is not a variable type. Variable types in terraform are:

string: A sequence of Unicode characters

number: A numeric value, either an integer or a floating-point number

bool: A Boolean value of either true or false

list: An ordered collection of values, all of which must be of the same type

map: A collection of key-value pairs, where the keys and values can be of any type

set: An unordered collection of unique values, all of which must be of the same type

object: A complex structure that can contain other objects, lists, and scalar values

tuple: A fixed-length array of specific types. (Terraform version 0.13 and later)

Reference:

https://developer.hashicorp.com/terraform/language/expressions/types

Question: 215 CertyIQ

What is a key benefit of the Terraform state file?

A. A state file can be used to schedule recurring infrastructure tasks
B. A state file represents a source of truth for resources provisioned with a public cloud console
C. A state file represents the desired state expressed by the Terraform code files
D. A state file represents a source of truth for resources provisioned with Terraform

Answer: D

Explanation:

D. A state file represents a source of truth for resources provisioned with Terraform

A key benefit of the Terraform state file is that it represents a source of truth for resources provisioned with Terraform. The state file is used to keep track of the current state of the infrastructure resources that are being managed by Terraform. It contains information about the resources, their properties, and the dependencies between them.

The state file is used by Terraform to determine what changes need to be made to the infrastructure to reach the desired state defined in the configuration files.

A. A state file can be used to schedule recurring infrastructure tasks, is not a benefit of state file.

B. A state file represents a source of truth for resources provisioned with a public cloud console, is not a benefit of state file.

C. A state file represents the desired state expressed by the Terraform code files, is not a benefit of state file.

Question: 216 CertyIQ

Which of these statements about Terraform Enterprise workspaces is false?

A.They can securely store cloud credentials

B.You must use the CLI to switch between workspaces

C.Plans and applies can be triggered via version control system integrations

D.They have role-based access controls

Answer: B

Explanation:

1. B. You must use the CLI to switch between workspacesThis statement is false. In Terraform Enterprise, workspaces are designed to be managed through the web UI, API, or version control system (VCS) integrations, rather than using the Terraform CLI to switch between them. Each workspace in Terraform Enterprise corresponds to a specific environment or configuration, and you can manage them using the available integrations, allowing for better collaboration and automation.
2. Correct answer: BThe statement that is false is that "You must use the CLI to switch between workspaces" in Terraform Enterprise.Terraform Enterprise provides a web-based UI that allows you to switch between workspaces, view the state of your infrastructure, and run Terraform commands without having to use the command line interface.

Question: 217 CertyIQ

Define the purpose of state in Terraform.

A.State is used to map real world resources to your configuration and keep track of metadata

B.State is a method of codifying the dependencies of related resources

C.State is used to enforce resource configurations that relate to compliance policies

D.State is used to store variables and quickly reuse existing code

Answer: A

Explanation:

Textbook Ans: 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.

https://developer.hashicorp.com/terraform/language/state

Question: 218 CertyIQ

Which backend does the Terraform CLI use by default?

A.API

B.Remote

C.Terraform Cloud

D.Local

E.HTTP

Answer: D

Explanation:

D. Local By default, the Terraform CLI uses the local backend to store the state file. The local backend stores the state as a file on your local filesystem. If you want to use a different backend, such as a remote backend (e.g., Terraform Cloud) or other options like S3 or Consul, you need to configure the backend explicitly in your Terraform configuration files.

Question: 219 CertyIQ

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

A.True
B.False

Answer: B

Explanation:

B. False Using the terraform state r m command does not destroy the actual resource. Instead, it removes the resource's entry from the Terraform state file. As a result, Terraform will no longer manage or track the resource. However, the resource itself will still exist in your infrastructure. If you want to destroy the resource, you should use the terraform destroy command or modify your configuration to remove the resource and run terraform apply.

Question: 220 CertyIQ

Which method for sharing Terraform configurations keeps them confidential within your organization, supports Terraform's semantic version constraints, and provides a browsable directory?

A.Generic git repository
B.Terraform Cloud/Terraform Enterprise private module registry
C.Public Terraform Module Registry
D.Subfolder within a workspace

Answer: B

Explanation:

The Terraform Cloud/Terraform Enterprise private module registry allows organizations to store and share Terraform configurations in a private and secure way. This method keeps the configurations confidential within your organization, supports Terra form's semantic version constraints, and provides a browsable directory.

Question: 221 CertyIQ

You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP

address returned in the root configuration. You name the instance resource “main”.

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

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

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

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

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

Answer: A

Explanation:

1. A is the new one but both would work:output "name-rg" value = azurerm_resource_group.example.name output "name-rg2" value = "$ azurerm_resource_group.example.name " Changes to Outputs: + name-rg = "example-resource-group" + name-rg2 = "example-resource-group"
2. A for sure

Question: 222 CertyIQ

How would you refer to the indexing instance from the below configuration?

```
resource "aws_instance" "web" {
  ...
  for_each = {
    "terraform": "value1",
    "resource": "value2",
    "indexing": "value3",
    "example": "value4",
  }
}
```

A.aws_instance[“web”][“indexing”]
B.aws_instance.web.indexing
C.aws_instance-web[“indexing”]
D.aws_instance.web[“indexing”]

Answer: D

Explanation:

D. aws_ instance. web[“indexing”]To refer to the “indexing” instance in the given configuration, you would use the resource type followed by the resource name, and then use the index key in square brackets. The correct syntax is: a ws_ instance. web["indexing"]This refers to the “indexing” instance created by the aw s _instance resource named “web”. The other options provided do not use the correct syntax for referencing resources with for_ each keys in Terraform.

Question: 223 CertyIQ

Which feature is not included in Terraform Cloud’s free tier?

A.Workspace
B.Remote state management
C.Audit logging
D.Private module registry

Answer: C

Explanation:

Audit logging is not included in Terraform Cloud's free tier. Terraform Cloud is a hosted service for Terraform that provides remote state management, a private module registry, and collaboration features such as workspaces and version control integration. However, audit logging is not included in the free tier

https://www.hashicorp.com/products/terraform/pricingAudit logging only for Cloud Business and Enterprise not for Cloud Free Cloud Team & Governance in 2023

Question: 224 CertyIQ

When should you run terraform init?

A.After you run terraform apply for the first time in a new Terraform project and before you run terraform plan
B.After you run terraform plan for the first time in a new Terraform project and before you run terraform apply
C.After you start coding a new Terraform project and before you run terraform plan for the first time
D.Before you start coding a new Terraform project

Answer: C

Explanation:

1. C.his is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control.
2. After you start coding a new Terraform project and before you run Terraform plan for the first time

Question: 225 CertyIQ

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

A.True

B.False

Answer: B

Explanation:

1. provider "aws" region = "us-west-2" access_key = "ACCESS_KEY" secret_key = "SECRET_KEY" provider "google" project = "my-gcp-project" credentials = "path/to/google/credentials.json"
2. B. FalseA Terraform configuration can contain multiple provider types. Providers in Terraform are responsible for understanding API interactions and exposing resources for various cloud platforms and infrastructure services. You can use multiple providers in a single Terraform configuration to manage resources across different platforms or services.

Question: 226 CertyIQ

You are making changes to existing Terraform code to add some new infrastructure.

When is the best time to run terraform validate?

A.After you run terraform plan so you can validate that your state file is consistent with your infrastructure

B.Before you run terraform plan so you can validate your code syntax

C.Before you run terraform apply so you can validate your infrastructure changes

D.After you run terraform apply so you can validate that your infrastructure is reflected in your code

Answer: B

Explanation:

B. Before you run terraform plan so you can validate your code syntax. Running terraform validate before running terraform plan helps catch syntax errors, missing or incorrect argument declarations, and other issues before any changes are made to the infrastructure.

Question: 227 CertyIQ

How does Terraform manage most dependencies between resources?

A.By defining dependencies as modules and including them in a particular order

B.The order that resources appear in Terraform configuration indicates dependencies

C.Using the depends_on parameter

D.Terraform will automatically manage most resource dependencies

Answer: D

Explanation:

D. Terraform will automatically manage most resource dependencies. Option A is incorrect because Terraform does not use modules to manage dependencies. Option B is incorrect because the order that resources appear

in Terraform configuration does not indicate dependencies. Option C is incorrect because the depends_on parameter is used to explicitly declare dependencies, not to automatically manage them.

Question: 228 CertyIQ

What does running a terraform plan do?

A.Imports all of your existing cloud provider resources to the state file

B.Compares the state file to your Terraform code and determines if any changes need to be made

C.Imports all of your existing cloud provider resources to your Terraform configuration file

D.Compares your Terraform code and local state file to the remote state file in a cloud provider and determines if any changes need to be made

Answer: B

Explanation:

1. B. Compares the state file to your Terraform code and determines if any changes need to be made. The terraform plan command is used to create an execution plan, which shows you the changes that will be made to your infrastructure based on the current Terraform configuration and the current state file. It allows you to review the changes before actually applying them, helping you understand the impact of your changes and catch any unintended modifications before they happen.
2. B. Compares the state file to your Terraform code and determines if any changes need to be madeRunning a terraform plan command performs a comparison between your Terraform configuration files and the current state file. It then generates an execution plan that shows the differences between the desired state (defined in the configuration) and the actual state (represented by the state file). This plan outlines the actions (create, update, or delete) that Terraform will take to reconcile the differences and achieve the desired state when the terraform apply command is executed.

Question: 229 CertyIQ

What are some benefits of using Sentinel with Terraform Cloud/Terraform Enterprise? (Choose three.)

A.Policy-as-code can enforce security best practices

B.You can restrict specific configurations on resources like "CIDR=0.0.0.0/0" not allowed

C.You can enforce a list of approved AWS AMIs

D.Sentinel Policies can be written in HashiCorp Configuration Language (HCL)

E.You can check out and check in cloud access keys

Answer: ABC

Explanation:

A. Policy-as-code can enforce security best practices B. You can restrict specific configurations on resources like "CIDR=0.0.0.0/0" not allowed C. You can enforce a list of approved AWS AMIs Sentinel is a policy-as-code framework that integrates with Terraform Cloud and Terraform Enterprise, allowing you to enforce policies on your infrastructure as part of the provisioning process. By using Sentinel, you can enforce security best practices, restrict specific configurations such as disallowing overly permissive CIDR blocks, and maintain a list of approved AWS AMIs, among other things. This helps to ensure that your infrastructure is secure, compliant, and adheres to organizational standards.

Question: 230 CertyIQ

You want to share Terraform state with your team, store it securely, and provide state locking.

How would you do this? (Choose three.)

A.Using the remote Terraform backend with Terraform Cloud / Terraform Enterprise.
B.Using the local backend.
C.Using the s3 terraform backend. The dynamodb_field option is not needed.
D.Using an s3 terraform backend with an appropriate IAM policy and dynamodb_field option configured.
E.Using the consul Terraform backend.

Answer: ADE

Explanation:

A. Using the remote Terraform backend with Terraform Cloud / Terraform Enterprise.This option allows you to store your Terraform state file securely in a managed service provided by HashiCorp. Terraform Cloud and Terraform Enterprise offer additional features such as access controls, remote execution, and collaboration tools.D. Using an s3 terraform backend with an appropriate IAM policy and dynamodb_field option configured.This option enables you to store your Terraform state file in an Amazon S3 bucket, which provides secure storage with access controls. By configuring an appropriate IAM policy, you can ensure that only authorized users and services can access the state file. Additionally, using the dynamodb_ field option allows you to enable state file locking with Amazon DynamoDB, preventing multiple users from concurrently modifying the state.E. Using the consul Terraform backend.This option allows you to store your Terraform state file in a Consul cluster, a distributed key-value store. Consul provides features like access control lists (ACLs) and TLS encryption to help you secure your state file.

Question: 231 CertyIQ

From which of these sources can Terraform import modules?

A.Local path
B.GitHub Repository
C.Terraform Module Registry
D.All of the above

Answer: D

Explanation:

D. All of the above Terraform can import modules from a variety of sources, including: A. Local path: You can reference a local directory containing your module's Terraform files. B. GitHub Repository: You can reference a GitHub repository, specifying the repository URL.C. Terraform Module Registry: You can import modules directly from the public Terraform Registry or from private module registries in Terraform Cloud or Terraform Enterprise. These options make it easy to reuse and share modules across different projects and teams.

Question: 232 CertyIQ

How would you output returned values from a child module?

A.Declare the output in the root configuration

B.Declare the output in the child module

C.Declare the output in both the root and child module

D.None of the above

Answer: B

Explanation:

1. b - https://developer.hashicorp.com/terraform/language/values/outputs
2. How would you output returned values from a child module?A. Declare the output in the root configurationB. Declare the output in the child moduleC. Declare the output in both the root and child moduleD. None of the above

Question: 233 CertyIQ

You have decided to create a new Terraform workspace to deploy a development environment.

What is different about this workspace?

A.It has its own state file

B.It pulls in a different terraform.tfvars file

C.It uses a different branch of code

D.It uses a different backend

Answer: A

Explanation:

A. It has its own state file When you create a new Terraform workspace, it has its own separate state file. This allows you to manage and deploy multiple environments with different configurations without interfering with each other. Each workspace's state file is isolated from the others, ensuring that changes in one environment don't affect the resources in another environment. While it is possible for different workspaces to use different terraform. tf Wars files, branches of code, or even different backends, these differences are not inherently tied to the creation of a new workspace. These variations would depend on how you choose to structure and manage your Terraform configurations for different environments.

Question: 234 CertyIQ

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

A.True

B.False

Answer: A

Explanation:

A. True Any user can publish modules to the public Terraform Module Registry. By publishing a module to the public registry, users can share their infrastructure code with the community and contribute to best practices. The public Terraform Module Registry contains a wide range of modules created by the community and Hashi Corp, which can be used to simplify Terraform configurations and promote code reusability.

Question: 235 CertyIQ

Which of these commands makes your code more human readable?

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

Answer: D

Explanation:
1. The correct answer is D. terraform fmt is used to format the Terraform configuration files, making them more human-readable and consistent. This command applies a standard formatting style to your code, which helps to reduce errors and makes it easier to read and maintain.terraform validate is used to check the syntax and validate the Terraform code for errors.terraform plan is used to generate an execution plan that describes what Terraform will do when you apply the configuration.terraform output is used to retrieve the values of output variables defined in your Terraform configuration.
2. It's D. terraform fmt

Question: 236 CertyIQ

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

A.True
B.False

Answer: A

Explanation:

True is a correct answer.

Question: 237 CertyIQ

Select the command that doesn't cause Terraform to refresh its state.

A.terraform apply
B.terraform destroy
C.terraform plan
D.terraform state list

Answer: D

Explanation:

D. terraform state list

Question: 238 CertyIQ

Sentinel policy-as-code is available in Terraform Enterprise.

A.True
B.False

Answer: A

Explanation:

Reference:

https://docs.hashicorp.com/sentinel/terraform

Question: 239 CertyIQ

Before you can use Terraform's remote backend, you must first execute terraform init.

A.True
B.False

Answer: A

Explanation:

This command initializes a working directory containing Terraform configuration files, downloads provider plugins and sets up the backend for state storage. If the configuration files have changed, terraform in it will also perform an update of the required provider plugins.

Question: 240 CertyIQ

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

A.Plan
B.Apply
C.Import
D.Init
E.Validate

Answer: BD

Explanation:

B .Apply

D .Init

Question: 241 CertyIQ

You are working on some new application features and you want to spin up a copy of your production deployment

to perform some quick tests. In order to avoid having to configure a new state backend, what open source Terraform feature would allow you create multiple states but still be associated with your current code?

A.Terraform data sources
B.Terraform local values
C.Terraform modules
D.Terraform workspaces
E.None of the above

Answer: D

Explanation:

D. Terraform workspaces

Question: 242 CertyIQ

Which provisioner invokes a process on the machine running Terraform?

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

Answer: C

Explanation:

https://developer.hashicorp.com/terraform/language/resources/provisioners/local-exec

Question: 243 CertyIQ

_____ backends support state locking.

A.Some
B.No
C.Only local
D.All

Answer: A

Explanation:

Reference:

https://developer.hashicorp.com/terraform/language/state/backends

Question: 244 CertyIQ

Which of the following methods, used to provision resources into a public cloud, demonstrates the concept of

infrastructure as code?

A.curl commands manually run from a terminal

B.A sequence of REST requests you pass to a public cloud API endpoint

C.A script that contains a series of public cloud CLI commands

D.A series of commands you enter into a public cloud console

Answer: B

Explanation:

1. B makes more sense
2. you declare your desired config and then Terraform performs a sequence of API calls to your cloud provider. It's not a script with CLI commands.

Question: 245 CertyIQ

Which of the following should you put into the required_providers block?

A.version >= 3.1

B.version = ">= 3.1"

C.version ~> 3.1

Answer: B

Explanation:

1. B since this syntax is correct. If you go back to the previous questions you'll see a screenshot where a relevant question is being asked.
2. Bhttps://developer.hashicorp.com/terraform/language/providers/requirements

Question: 246 CertyIQ

When should you write Terraform configuration files for existing infrastructure that you want to start managing with Terraform?

A.Before you run terraform import

B.You can import infrastructure without corresponding Terraform code

C.Terraform will generate the corresponding configuration files for you

D.After you run terraform import

Answer: A

Explanation:

Correct answer is A: Before you run terraform import

Question: 247 CertyIQ

Which command should you run to check if all code in a Terraform configuration that references multiple modules is properly formatted without making changes?

A.terraform fmt -write=false

B.terraform fmt -list -recursive

C.terraform fmt -check -recursive

D.terraform fmt -check

Answer: C

Explanation:

C. terraform fmt -check -recursive The terraform fmt -check -recursive command checks if the files are formatted according to the Terraform language style conventions. The -check option will make the command return a non-zero exit code if any of the files are not properly formatted, and the -recursive option instructs it to go into the sub-directories as well, which is useful when you have a Terraform configuration that references multiple modules.

Question: 248 CertyIQ

What features stops multiple users from operating on the Terraform state at the same time?

A.Provider constraints

B.Remote backends

C.State locking

D.Version control

Answer: C

Explanation:

C. State locking

Question: 249 CertyIQ

You are creating a reusable Terraform configuration and want to include a billing_dept tag so your Finance team can track team-specific spending on resources. Which of the following billing_dept variable declarations will allow you to do this?

A.
```
variable "billing_dept" {
    optional = true
}
```

B.
```
variable "billing_dept" {
    type = optional(string)
}
```

C.
```
variable "billing_dept" {
    default = ""
}
```

```
variable "billing_dept" {
    type = default
}
```

D.

Answer: C

Explanation:

C is the correct option. All the arguments of variables are optional. A - optional is not a valid argument. B- optional(string) is not a valid argument value. D - default is not a valid argument value.

Question: 250 CertyIQ

Which of these are secure options for storing secrets for connecting to a Terraform remote backend? (Choose two.)

A.Inside the backend block within the Terraform configuration

B.Defined in Environment variables

C.Defined in a connection configuration outside of Terraform

D.A variable file

Answer: BC

Explanation:

B .Defined in Environment variables

C. Defined in a connection configuration outside of Terraform

Question: 251 CertyIQ

You want to define a single input variable to capture configuration values for a server. The values must represent memory as a number, and the server name as a string.

Which variable type could you use for this input?

A.List

B.Object

C.Map

D.Terraform does not support complex input variables of different types

Answer: B

Explanation:

B. ObjectThe Object type in Terraform allows you to create complex input variables that contain more than one value and can be of different types.variable "server_config" type = object(memory = numbername = string) description = "Server configuration values" In this example, server_config is an object that expects two attributes: memory (a number) and name (a string).

Question: 252 CertyIQ

What does Terraform not reference when running a terraform apply -refresh-only?

A.Credentials
B.State file
C.Terraform resource definitions in configuration files
D.Cloud provider

Answer: C

Explanation:

The -refresh-only flag tells Terraform to skip the resource creation and update steps and only update the state file with the current state of resources in the cloud provider. This means that Terraform will not reference the resource definitions in the configuration files, since it is not applying any changes to the cloud provider based on those definitions.

Question: 253 CertyIQ

Multiple team members are collaborating on infrastructure using Terraform and want to format their Terraform code following standard Terraform-style convention. How could they automatically ensure the code satisfies conventions?

A.Run the terraform fmt command during the code linting phase of your CI/CD process
B.Manually apply two spaces indentation and align equal sign "=" characters in every Terraform file (*.tf)
C.Run the terraform validate command prior to executing terraform plan or terraform apply

Answer: A

Explanation:

By running terraform fmt during the code linting phase of your CI/CD process, you can automatically ensure that all Terraform files are formatted according to the standard conventions before any code is merged into the main branch. This helps to maintain a consistent code style and reduces the likelihood of errors caused by inconsistent formatting.

Question: 254 CertyIQ

When using a remote backend or Terraform Cloud integration, where does Terraform save resource state?

A.On the disk
B.In memory
C.In an environment variable
D.In the remote backend or Terraform Cloud

Answer: D

Explanation:

D. In the remote backend or Terraform Cloud With a remote backend configuration or Terraform Cloud integration, the state data, which includes information about the resources managed by Terraform, is stored

and managed in the remote backend or Terraform Cloud. This allows for centralized state management, collaboration, and concurrent access to the state file by multiple users or team members.

Question: 255 CertyIQ

In Terraform HCL, an object type of object(name=string, age=number) would match this value:

A.
```
{
  name = "John"
  age = fifty two
}
```

B.
```
{
  name = "John"
  age = 52
}
```

C.
```
{
  name =  John
  age = fifty two
}
```

D.
```
{
  name =  John
  age = 52
}
```

Answer: B

Explanation:

B is a correct answer.

Question: 256 CertyIQ

You add a new resource to an existing Terraform configuration, but do not update the version constraint in the configuration. The existing and new resources use the same provider. The working directory contains a .terraform-lock.hcl file.

How will Terraform choose which version of the provider to use?

A.Terraform will use the latest version of the provider for the new resource and the version recorded in the lock file to manage existing resources

B.Terraform will use the version recorded in your lock file

C.Terraform will check your state file to determine the provider version to use

D.Terraform will use the latest version of the provider available at the time you provision your new resource

Answer: B

Explanation:

Reference:

https://developer.hashicorp.com/terraform/language/files/dependency-lock#dependency-installation-behavior

Question: 257 CertyIQ

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

A.True

B.False

Answer: B

Explanation:

B. False the commands to use to interact with Terraform itself (such as terraform apply, terraform plan, and so on) remain the same regardless of the cloud provider.

Question: 258 CertyIQ

Define the purpose of state in Terraform.

A.State stores variables and lets you quickly reuse existing code

B.State lets you enforce resource configurations that relate to compliance policies

C.State codifies the dependencies of related resources

D.State maps real world resources to your configuration and keeps track of metadata

Answer: D

Explanation:

https://developer.hashicorp.com/terraform/language/state/purpose#mapping-to-the-real-worldhttps://developer.hashicorp.com/terraform/language/state/purpose#metadata

Question: 259 CertyIQ

Which of these actions will prevent two Terraform runs from changing the same state file at the same time?

A.Refresh the state after running Terraform
B.Delete the state before running Terraform
C.Configure state locking for your state backend
D.Run Terraform with parallelism set to 1

Answer: C

Explanation:

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

"https://developer.hashicorp.com/terraform/language/state/locking

Question: 260 CertyIQ

While attempting to deploy resources into your cloud provider using Terraform, you begin to see some odd behavior and experience slow responses. In order to troubleshoot you decide to turn on Terraform debugging. Which environment variables must be configured to make Terraform's logging more verbose?

A.TF_LOG_PATH
B.TF_VAR_log_level
C.TF_LOG
D.TF_VAR_log_path

Answer: C

Explanation:

"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.

"https://developer.hashicorp.com/terraform/internals/debugging

Question: 261 CertyIQ

The Terraform binary version and provider versions must match each other in a single configuration.

A.True
B.False

Answer: B

Explanation:

If the versions are incompatible or if a required provider is missing, Terraform will prompt you to update the provider versions or download the necessary provider plugins.

Question: 262 CertyIQ

The .terraform.lock.hcl file tracks module versions.

A.True
B.False

Answer: B

Explanation:

The .terraform. lock. hcl file is not used to track module versions; it is used to lock the versions of the provider dependencies used by your Terraform configuration.

Question: 263 CertyIQ

You can develop a custom provider to manage its resources using Terraform.

A.True
B.False

Answer: A

Explanation:

https://developer.hashicorp.com/terraform/tutorials/providers-plugin-framework/providers-plugin-framework-provider

Question: 264 CertyIQ

Which of these is not a benefit of remote state?

A.Keeping unencrypted sensitive information off disk
B.Easily share reusable code modules
C.Working in a team
D.Delegate output to other teams

Answer: D

Explanation:

The answer is D. Delegate output to other teams.Remote state is a feature of Terraform that allows you to store the state of your infrastructure in a remote location. This can be beneficial for a number of reasons,

including:Keeping unencrypted sensitive information off diskEasily sharing reusable code modulesWorking in a teamEnabling version control and auditingHowever, remote state does not allow you to delegate output to other teams. This is because the state file contains the full state of your infrastructure, including all of the resources that have been created. If you want to delegate output to other teams, you will need to use a different mechanism, such as a configuration management tool or a messaging system.

Question: 265 CertyIQ

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

A.depends_on

B.alias

C.id

D.name

Answer: B

Explanation:

Reference:

https://developer.hashicorp.com/terraform/language/providers/configuration#alias-multiple-provider-configurations

Question: 266 CertyIQ

A developer accidentally launched a VM (virtual machine) outside of the Terraform workflow and ended up with two servers with the same name. They don't know which VM Terraform manages but do have a list of all active VM IDs.

Which of the following methods could you use to discover which instance Terraform manages?

A.Run terraform taint/code on all the VMs to recreate them

B.Update the code to include outputs for the ID of all VMs, then run terraform plan to view the outputs

C.Run terraform state list to find the names of all VMs, then run terraform state show for each of them to find which VM ID Terraform manages

D.Use terraform refresh/code to find out which IDs are already part of state

Answer: C

Explanation:

Reference:

https://developer.hashicorp.com/terraform/cli/commands/state/listhttps://developer.hashicorp.

com/terraform/cli/commands/state/show

Question: 267 CertyIQ

Which of the following is not considered a safe way to inject sensitive values into a Terraform Cloud workspace?

A.Edit the state file directly just before running terraform apply
B.Set the variable value on the command line with the -var flag
C.Write the value to a file and specify the file with the -var-file flag

Answer: A

Explanation:

Choosing A as it is not right way to do it. Even though C is not preferred, I think you can do it as you are supplying it as a parameter during run and the file is local to the operator running the command and hence can delete it after that. Since we are talking about state being stored remotely, it works very much similar to option B.

Question: 268 CertyIQ

If you update the version constraint in your Terraform configuration, Terraform will update your lock file the next time you run terraform init.

A.True
B.False

Answer: B

Explanation:

1. You will get an error. You have to run terraform init -upgrade to change the version in .terraform.lock.hcl file.
2. While I chose B (false), the question is not clear. Reason , it will not automatically update the version with terraform init, but there is an option -upgrade that needs to be added (terraform init -upgrade). Not sure if the exam has an implicit assumption that you run `terraform init ` with -upgrade option and if so the answer would be true.

Question: 269 CertyIQ

You must initialize your working directory before running terraform validate.

A.True
B.False

Answer: A

Explanation:

"Validation requires an initialized working directory with any referenced plugins and modules installed. "https://developer.hashicorp.com/terraform/cli/commands/validate

Question: 270 CertyIQ

If you manually destroy infrastructure, what is the best practice reflecting this change in Terraform?

A.Manually update the state fire

B.Remove the resource definition from your file and run terraform apply -refresh-only

C.Run terraform import

D.It will happen automatically

Answer: B

Explanation:

B. Remove the resource definition from your file and run terraform apply -refresh-only

Question: 271 CertyIQ

You created infrastructure outside of the Terraform workflow that you now want to manage using Terraform. Which command brings the infrastructure into Terraform state?

A.terraform init

B.terraform get

C.terraform refresh

D.terraform import

Answer: D

Explanation:

https://developer.hashicorp.com/terraform/cli/import

Question: 272 CertyIQ

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

A.When you change a Terraform-managed resource via the Azure Cloud Console, Terraform updates the state file to reflect the change during the next plan or apply

B.Changing resources via the Azure Cloud Console records the change in the current state file

C.When you change a resource via the Azure Cloud Console, Terraform records the changes in a new state file

D.Changing resources via the Azure Cloud Console does not update current state file

Answer: D

Explanation:

1. A. It is not correct, terraform plan doesn't reflect the changes, it only shows the changes to you when you run terraform apply it reflects the change.B. When you do manual change in resource, Terraform doesn't imply changes automaticallyC. Terraform never creates a new state file for the new changes and it uses the existing one unless you don't specify a different backend.D. Correct. You can only update tfstate file with either terraform apply (suggested) or terraform apply -refresh-only (not suggested though).Option A has probably typo issue, terraform plan shouldn't be in there.

Question: 273 CertyIQ

Which statement describes a goal of infrastructure as code?

A.A pipeline process to test and deliver software
B.Defining a vendor-agnostic API
C.Write once, run anywhere
D.The programmatic configuration of resources

Answer: D

Explanation:

The programmatic configuration of resources

Question: 274 CertyIQ

terraform validate confirms the syntax of Terraform files.

A.True
B.False

Answer: A

Explanation:

A. True Correct, terraform validate is used to confirm the syntax and structure of Terraform files. It checks whether your configuration files are correctly written and can be successfully parsed by Terraform. This helps catch simple errors or typos in your configuration before you proceed with other Terraform commands like terraform plan or terraform apply.

Question: 275 CertyIQ

Which command adds existing resources into Terraform state?

A.terraform init
B.terraform plan
C.terraform refresh
D.terraform import
E.All of these

Answer: D

Explanation:

D: terraform import

Question: 276 CertyIQ

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

A.True
B.False

Answer: B

Explanation:

B. False Never save secret data in version control

Question: 277 CertyIQ

Which of the following commands would you use to access all of the attributes and details of a resource managed by Terraform?

A.terraform state list 'provider_type.name'
B.terraform state show 'provider_type.name'
C.terraform get 'provider_type.name'
D.terraform state list

Answer: B

Explanation:

B. terraform state show 'provider_type.name'

Question: 278 CertyIQ

terraform validate confirms that your infrastructure matches the Terraform state file.

A.True
B.False

Answer: B

Explanation:

B. False The statement is not accurate. terraform validate is a command in Terraform that checks whether the configuration files are syntactically and structurally correct. It validates the configuration syntax and structure but does not confirm that your infrastructure matches the Terraform state file. To confirm that your infrastructure matches the Terraform state file, you would typically use commands like terraform plan to compare the current state of your configuration with the state recorded in the Terraform state file. This helps you understand the differences between the desired configuration and the actual infrastructure that Terraform is managing.

Question: 279 CertyIQ

A senior 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
D.Stop and generate an error message about the missing instances

Answer: C

Explanation:

C. Rebuild only the instances that were deleted When you run terraform apply after some instances have been accidentally deleted by a senior admin, Terraform will detect the differences between your desired configuration (defined in your Terraform files) and the actual state of your cloud infrastructure (as recorded in the Terraform state file). It will then take the necessary actions to reconcile the two. In this case, Terraform will identify that the instances were deleted and no longer exist in the state file. It will then create new instances to match the desired configuration, effectively rebuilding only the instances that were deleted. This approach is one of the key benefits of using Terraform: it helps maintain the desired infrastructure state even when changes occur outside of Terra form's control.

Question: 280 CertyIQ

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

A.True
B.False

Answer: B

Explanation:

The terraform in it command does not create an example main.tf file in the current directory. Its primary purpose is to initialize the working directory for Terraform configuration and set up the backend, providers, and modules specified in your configuration. It doesn't generate or modify any configuration files; you need to create your main.tf file yourself to define your infrastructure resources and settings.

Question: 281 CertyIQ

Which argument helps prevent unexpected updates when calling Terraform Registry modules?

A.count
B.source
C.version
D.lifecycle

Answer: C

Explanation:

C. version The version argument helps prevent unexpected updates when calling Terraform Registry modules. It allows you to specify the exact version of the module you want to use. This ensures that your configuration consistently uses the same version of the module until you explicitly decide to update it to a newer version. Using the version argument is a good practice to ensure that your infrastructure remains stable and predictable, as it prevents unintentional changes caused by automatic module updates.

Question: 282 CertyIQ

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

A.True
B.False

Answer: B

Explanation:

B Because it's stdrr

https://developer.hashicorp.com/terraform/internals/debugging

Question: 283 CertyIQ

How would you output returned values from a child module in the Terraform CLI output?

A.Declare the output in the root configuration
B.Declare the output in the child module
C.Declare the output in both the root and child module
D.None of the above

Answer: C

Explanation:

to output returned values from a child module to CLI - Declare the output in both the root and child module. For example:child-module.tfoutput "child_foo" value = "foobar" main.tfmodule "child" source = "path/to/child" output "output_to_cli" value = module.child.child_foo

Thank you

Thank you for being so interested in the premium exam material.
I'm glad to hear that you found it informative and helpful.

If you have any feedback or thoughts on the bumps, I would love to hear them.
Your insights can help me improve our writing and better understand our readers.

Best of Luck

You have worked hard to get to this point, and you are well-prepared for the exam
Keep your head up, stay positive, and go show that exam what you're made of!

Feedback

More Papers