

TA-002-P Dumps

HashiCorp Certified: Terraform Associate

<https://www.certleader.com/TA-002-P-dumps.html>



NEW QUESTION 1

- (Exam Topic 1)

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:

Somewhat ambiguous question however the key phrase is "feature". You need a remote backend first with a State Locking feature available to avoid this scenario.
<https://blog.gruntwork.io/how-to-manage-terraform-state-28f5697e68fa>

NEW QUESTION 2

- (Exam Topic 1)

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

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

Answer: C

Explanation:

terraform workspace list

The command will list all existing workspaces.

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

NEW QUESTION 3

- (Exam Topic 1)

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:

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

NEW QUESTION 4

- (Exam Topic 1)

When should you use the force-unlock command?

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

Answer: C

Explanation:

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

NEW QUESTION 5

- (Exam Topic 1)

Terraform provisioners can be added to any resource block.

- A. True
- B. False

Answer: A

Explanation:

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

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

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

NEW QUESTION 6

- (Exam Topic 1)

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>

NEW QUESTION 7

- (Exam Topic 1)

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 8

- (Exam Topic 1)

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>

NEW QUESTION 9

- (Exam Topic 1)

FILL BLANK

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

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

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

State

JUMP TO SECTION ▾

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

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

NEW QUESTION 10

- (Exam Topic 1)

Which task does terraform init not perform?

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

Answer: D

Explanation:

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

NEW QUESTION 10

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 12

- (Exam Topic 1)

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

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

Answer: C

Explanation:

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

NEW QUESTION 15

- (Exam Topic 1)

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:

"Overuse of dynamic blocks can make configuration hard to read and maintain, so we recommend using them only when you need to hide details in order to build a clean user interface for a re-usable module. Always write nested blocks out literally where possible."

Reference: <https://github.com/hashicorp/terraform/issues/19291>

NEW QUESTION 19

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 22

- (Exam Topic 1)

Only the user that generated a plan may apply it.

- A. True
- B. False

Answer: B

NEW QUESTION 25

- (Exam Topic 1)

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

NEW QUESTION 30

- (Exam Topic 1)

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

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

Answer: A

Explanation:

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

NEW QUESTION 35

- (Exam Topic 1)

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

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

Answer: D

Explanation:

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

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

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

NEW QUESTION 39

- (Exam Topic 1)

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

- A. True
- B. False

Answer: A

Explanation:

Output values are like function return values.

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

NEW QUESTION 40

- (Exam Topic 1)

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

- A. True
- B. False

Answer: B

NEW QUESTION 44

- (Exam Topic 1)

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

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

Answer: B

Explanation:

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

NEW QUESTION 47

- (Exam Topic 1)

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

- A. True
- B. False

Answer: A

Explanation:

Most provisioners require access to the remote resource via SSH or WinRM, and expect a nested connection block with details about how to connect. Reference: <https://www.terraform.io/docs/language/resources/provisioners/connection.html>

NEW QUESTION 50

- (Exam Topic 1)

What does the default "local" Terraform backend store?

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

Answer: D

Explanation:

The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally. Reference: <https://www.terraform.io/docs/language/settings/backends/local.html>

NEW QUESTION 55

- (Exam Topic 1)

A Terraform provider is not responsible for:

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

Answer: B

Explanation:

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

NEW QUESTION 56

- (Exam Topic 1)

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:

"The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration. Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped. While this may seem tedious, it still gives Terraform users an avenue for importing existing resources."
<https://www.terraform.io/cli/import/usage>

NEW QUESTION 60

- (Exam Topic 2)

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

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

Answer: D

Explanation:

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

NEW QUESTION 62

- (Exam Topic 2)

While using generic git repository as a module source, which of the below options allows terraform to select a specific version or tag instead of selecting the HEAD.

- A. Append ref argument asmodule "vpc" { source = "git::https://example.com/vpc.git?ref=v1.2.0"}
- B. Append version argument asmodule "vpc" { source = "git::https://example.com/vpc.git?version=v1.2.0"}
- C. Append ref argument asmodule "vpc" { source = "git::https://example.com/vpc.git#ref=v1.2.0"}
- D. By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository and you can not override this.

Answer: A

Explanation:

By default, Terraform will clone and use the default branch (referenced by HEAD) in the selected repository. You can override this using the ref argument:

```
module "vpc" {
source = "git::https://example.com/vpc.git?ref=v1.2.0"
}
```

The value of the ref argument can be any reference that would be accepted by the git checkout command, including branch and tag names.

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

NEW QUESTION 64

- (Exam Topic 2)

You want to use different AMI images for different regions and for the purpose you have defined following code block.

```
* 1.variable "images"
* 2.{
* 3. type = "map"
* 4.
* 5. default = {
* 6. us-east-1 = "image-1234"
* 7. us-west-2 = "image-4567"
* 8. us-west-1 = "image-4589"
* 9. }
* 10.}
```

What of the following approaches needs to be followed in order to select image-4589?

- A. var.images["us-west-1"]
- B. var.images[3]
- C. var.images[2]
- D. lookup(var.images["us-west-1"])

Answer: A

NEW QUESTION 66

- (Exam Topic 2)

Matt wants to import a manually created EC2 instance into terraform so that he can manage the EC2 instance through terraform going forward. He has written the configuration file of the EC2 instance before importing it to Terraform. Following is the code:

```
resource "aws_instance" "matt_ec2" { ami = "ami-bg2640de" instance_type = "t2.micro" vpc_security_group_ids = ["sg-6ae7d613", "sg-53370035"] key_name = "mysecret" subnet_id = "subnet-9e3cfbc5" }
```

The instance id of that EC2 instance is i-0260835eb7e9bd40 How he can import data of EC2 to state file?

- A. terraform import aws_instance.id = i-0260835eb7e9bd40
- B. terraform import i-0260835eb7e9bd40
- C. terraform import aws_instance.i-0260835eb7e9bd40

D. terraform import aws_instance.matt_ec2 i-0260835eb7e9bd40

Answer: D

Explanation:

<https://www.terraform.io/docs/import/usage.html>

NEW QUESTION 70

- (Exam Topic 2)

Provisioners should only be used as a last resort.

A. False

B. True

Answer: B

Explanation:

Provisioners are a Last Resort

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

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

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

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

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

NEW QUESTION 71

- (Exam Topic 2)

You want to use terraform import to start managing infrastructure that was not originally provisioned through infrastructure as code. Before you can import the resource's current state, what must you do in order to prepare to manage these resources using Terraform?

A. Run terraform refresh to ensure that the state file has the latest information for existing resources.

B. Update the configuration file to include the new resources.

C. Shut down or stop using the resources being imported so no changes are inadvertently missed.

D. Modify the Terraform state file to add the new resources.

Answer: B

Explanation:

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

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

The terraform import command is used to import existing infrastructure.

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

Example:

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

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

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

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

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

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

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 75

- (Exam Topic 2)

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

A. Terraform Backends

B. Terraform Functions

C. Dynamic Blocks

D. Terraform Expression

Answer: C

NEW QUESTION 78

- (Exam Topic 2)

What is the default backend for Terraform?

A. consul

B. gcs

- C. local
- D. etcd

Answer: C

Explanation:

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

NEW QUESTION 83

- (Exam Topic 2)

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

- A. False
- B. True

Answer: B

Explanation:

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

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

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

NEW QUESTION 84

- (Exam Topic 2)

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

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

Answer: C

Explanation:

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

NEW QUESTION 88

- (Exam Topic 2)

Workspaces in Terraform provides similar functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

- A. True
- B. False

Answer: B

Explanation:

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

Workspaces, managed with the terraform workspace command, aren't the same thing as Terraform Cloud's workspaces. Terraform Cloud workspaces act more like completely separate working directories; CLI workspaces are just alternate state files.

NEW QUESTION 92

- (Exam Topic 2)

terraform refresh command will not modify infrastructure, but does modify the state file.

- A. True
- B. False

Answer: A

Explanation:

The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. This can be used to detect any drift from the last-known state, and to update the state file. This does not modify infrastructure, but does modify the state file.

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

NEW QUESTION 95

- (Exam Topic 2)

Terraform works well in Windows but a Windows server is required.

- A. False
- B. True

Answer: A

Explanation:

You may see this QUESTION NO: in actual exam. Please remember : Terraform does not require GO language to be installed as a prerequisite and it does not require a Windows Server as well.

NEW QUESTION 100

- (Exam Topic 2)

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

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

Answer: C

Explanation:

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

NEW QUESTION 105

- (Exam Topic 2)

Which of the following best describes a Terraform provider?

- A. A plugin that Terraform uses to translate the API interactions with the service or provider.
- B. Serves as a parameter for a Terraform module that allows a module to be customized.
- C. Describes an infrastructure object, such as a virtual network, compute instance, or other components.
- D. A container for multiple resources that are used together.

Answer: A

Explanation:

A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. Alibaba Cloud, AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g. Terraform Cloud, DNSimple, Cloudflare).
<https://www.terraform.io/docs/providers/index.html>

NEW QUESTION 106

- (Exam Topic 2)

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

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

Answer: BE

Explanation:

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

NEW QUESTION 108

- (Exam Topic 2)

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

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

Answer: BC

Explanation:

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

NEW QUESTION 109

- (Exam Topic 2)

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

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

Answer: D

Explanation:

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

NEW QUESTION 113

- (Exam Topic 2)

Which of the following best describes the default local backend?

- A. The local backend is where Terraform Enterprise stores logs to be processed by an log collector.
- B. The local backend stores state on the local filesystem, locks the state using system APIs, and performs operations locally.
- C. The local backend is the directory where resources deployed by Terraform have direct access to in order to update their current state.
- D. The local backend is how Terraform connects to public cloud services, such as AWS, Azure, or GCP.

Answer: B

Explanation:

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

```
terraform { backend "local" {  
  path = "relative/path/to/terraform.tfstate"  
}  
}
```

<https://www.terraform.io/docs/backends/types/local.html>

NEW QUESTION 117

- (Exam Topic 3)

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

"\${var.instance_id}"

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

Answer: C

NEW QUESTION 121

- (Exam Topic 3)

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 124

- (Exam Topic 3)

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

- A. False
- B. True

Answer: A

Explanation:

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

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

NEW QUESTION 128

- (Exam Topic 3)

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

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

- C. terraform view
- D. terraform list

Answer: A

Explanation:

The terraform state list command is used to list resources within a Terraform state. Usage: terraform state list [options] [address...]
The command will list all resources in the state file matching the given addresses (if any). If no addresses are given, all resources are listed.
<https://www.terraform.io/docs/commands/state/list.html>

NEW QUESTION 133

- (Exam Topic 3)

The canonical format may change in minor ways between Terraform versions, so after upgrading Terraform it is recommended to proactively run.

- A. terraform fmt
- B. terraform init
- C. terraform validate
- D. terraform plan

Answer: A

NEW QUESTION 136

- (Exam Topic 3)

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

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

- A. YES
- B. No

Answer: B

Explanation:

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

NEW QUESTION 139

- (Exam Topic 3)

In regards to Terraform state file, select all the statements below which are correct?

- A. When using local state, the state file is stored in plain-text.
- B. The state file is always encrypted at rest.
- C. Storing state remotely can provide better security.
- D. Using the mask feature, you can instruct Terraform to mask sensitive data in the state file.
- E. The Terraform state can contain sensitive data, therefore the state file should be protected from unauthorized access.
- F. Terraform Cloud always encrypts state at rest.

Answer: ACEF

Explanation:

Terraform state can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.

When using local state, state is stored in plain-text JSON files.

When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.

Storing Terraform 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.

Recommendations

If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.

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.

NEW QUESTION 141

- (Exam Topic 3)

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

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

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

D. 1.2.0

Answer: AC

Explanation:

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

NEW QUESTION 142

- (Exam Topic 3)

What happens when a terraform apply command is executed?

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

Answer: B

Explanation:

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

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

NEW QUESTION 145

- (Exam Topic 3)

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

- A. required-provider
- B. required-version
- C. required_providers
- D. required_versions

Answer: C

Explanation:

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

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

NEW QUESTION 150

- (Exam Topic 3)

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

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

Answer: D

Explanation:

When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.

Terraform Cloud can also be used with local operations, in which case only state is stored in the Terraform Cloud backend.

<https://www.terraform.io/docs/backends/types/remote.html>

NEW QUESTION 153

- (Exam Topic 3)

The terraform state command can be used to _____

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

Answer: A

Explanation:

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

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

NEW QUESTION 158

- (Exam Topic 3)

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

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

Answer: A

Explanation:

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

NEW QUESTION 162

- (Exam Topic 3)

Why is it a good idea to declare the required version of a provider in a Terraform configuration file?

- * 1. terraform
- * 2. {
- * 3. required_providers
- * 4. {
- * 5. aws = "~> 1.0"
- * 6. }
- * 7. }

- A. To remove older versions of the provider.
- B. To ensure that the provider version matches the version of Terraform you are using.
- C. Providers are released on a separate schedule from Terraform itself; therefore a newer version could introduce breaking changes.
- D. To match the version number of your application being deployed via Terraform.

Answer: C

NEW QUESTION 164

- (Exam Topic 3)

Which of the following are string functions? Select three

- A. tostring
- B. tonumber
- C. Chomp
- D. format
- E. join

Answer: CDE

Explanation:

tonumber and tostring are Type Conversion function <https://www.terraform.io/docs/configuration/functions.html>

NEW QUESTION 167

- (Exam Topic 3)

Refer to the following terraform variable definition

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

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

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

Answer: B

NEW QUESTION 170

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 173

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

NEW QUESTION 178

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 183

- (Exam Topic 4)

True or False. The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. If drift is detected between the real-world infrastructure and the last known-state, it will modify the infrastructure to correct the drift.

- A. False
- B. True

Answer: A

Explanation:

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

NEW QUESTION 187

- (Exam Topic 4)

Where can Terraform not load a provider from?

- A. Plugins directory
- B. Provider plugin cache
- C. Official HashiCorp distribution on releases, hashicorp.com
- D. Source code

Answer: D

NEW QUESTION 190

- (Exam Topic 4)

Terraform console provides an interactive command-line console for evaluating and experimenting with expressions. You can use it to test interpolations before using them in configurations and to interact with any values currently saved in state.

Which configuration consistency errors does terraform validate report?

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

Answer: D

Explanation:

validate will look for syntax errors "Declaring a resource identifier more than once" is a syntax error

NEW QUESTION 193

- (Exam Topic 4)

Which of the following value will be accepted for `my_var`?

- * 1. variable "my_var"
- * 2. {
- * 3. type = string
- * 4. }

- A. 15
- B. "15"
- C. Both A and B
- D. None of the above

Answer: C

Explanation:

The Terraform language will automatically convert number and bool values to string values when needed, and vice-versa as long as the string contains a valid representation of a number or boolean value. Example

* true converts to "true", and vice-versa

* false converts to "false", and vice-versa

* 15 converts to "15", and vice-versa

Where possible, Terraform automatically converts values from one type to another in order to produce the expected type. If this isn't possible, Terraform will produce a type mismatch error and you must update the

configuration with a more suitable expression. <https://www.terraform.io/docs/configuration/expressions.html#type-conversion>

NEW QUESTION 196

- (Exam Topic 4)

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

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

Answer: AD

Explanation:

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

NEW QUESTION 198

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

NEW QUESTION 203

- (Exam Topic 4)

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

- A. True
- B. False

Answer: A

Explanation:

Initialization

Whenever a configuration's backend changes, you must run terraform init again to validate and configure the backend before you can perform any plans, applies, or state operations.

When changing backends, Terraform will give you the option to migrate your state to the new backend. This lets you adopt backends without losing any existing state.

To be extra careful, we always recommend manually backing up your state as well. You can do this by simply copying your terraform.tfstate file to another location. The initialization process should create a backup as well, but it never hurts to be safe!

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

NEW QUESTION 208

- (Exam Topic 4)

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

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

You immediately run terraform apply but don't see any changes. Your state file didn't move. Which command will migrate your current state file to the new S3 remote backend?

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

Answer: B

NEW QUESTION 210

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

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

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

NEW QUESTION 212

- (Exam Topic 4)

What feature of Terraform Cloud and/or Terraform Enterprise can you publish and maintain a set of custom modules which can be used within your organization?

- A. Terraform registry
- B. custom VCS integration
- C. private module registry
- D. remote runs

Answer: C

NEW QUESTION 214

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

NEW QUESTION 217

- (Exam Topic 4)

What does terraform import allow you to do?

- A. Import a new Terraform module
- B. Use a state file to import infrastructure to the cloud
- C. Import provisioned infrastructure to your state file
- D. Import an existing state file to a new Terraform workspace

Answer: C

NEW QUESTION 221

- (Exam Topic 4)

By default, where does Terraform store its state file?

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

Answer: D

Explanation:

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

NEW QUESTION 226

- (Exam Topic 4)

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

- A. name
- B. alias
- C. depends_on
- D. id

Answer: B

NEW QUESTION 230

- (Exam Topic 4)

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

- A. True
- B. False

Answer: B

Explanation:

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

NEW QUESTION 232

- (Exam Topic 4)

A terraform apply can not _____ infrastructure.

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

Answer: A

NEW QUESTION 235

- (Exam Topic 4)

The following is a snippet from a Terraform configuration file: Which, when validated, results in the following error: Fill in the blank in the error message with the correct string from the list below.

- A. version
- B. multi
- C. label
- D. alias

Answer: D

Explanation:

<https://www.terraform.io/docs/configuration/providers.html#alias-multiple-providerinstances>

NEW QUESTION 239

- (Exam Topic 4)

If a Terraform creation-time provisioner fails, what will occur by default?

- A. The resource will not be affected, but the provisioner will need to be applied again
- B. The resource will be destroyed
- C. The resource will be marked as "tainted"
- D. Nothing, provisioners will not show errors in the command line

Answer: C

Explanation:

If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply .

NEW QUESTION 244

- (Exam Topic 4)

You have written a terraform IaC script which was working till yesterday , but is giving some vague error from today , which you are unable to understand . You want more detailed logs that could potentially help you troubleshoot the issue , and understand the root cause. What can you do to enable this setting? Please note , you are using terraform OSS.

- A. Terraform OSS can push all its logs to a syslog endpoint
- B. As such, you have to set up the syslog sink, and enable TF_LOG_PATH env variable to the syslog endpoint and all logs will automatically start streaming.
- C. Detailed logs are not available in terraform OSS, except the crash messag
- D. You need to upgrade to terraform enterprise for this point.
- E. Enable the TF_LOG_PATH to the log sink file location, and logging output will automatically be stored there.
- F. Enable TF_LOG to the log level DEBUG, and then set TF_LOG_PATH to the log sink file location.Terraform debug logs will be dumped to the sink path, even in terraform OSS.

Answer: D

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 TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is

the default if TF_LOG is set to something other than a log level name.

To persist logged output you can set TF_LOG_PATH in order to force the log to always be appended to a specific file when logging is enabled. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

NEW QUESTION 245

- (Exam Topic 4)

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

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

Answer: A

Explanation:

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

NEW QUESTION 247

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 248

- (Exam Topic 4)

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

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

Answer: B

NEW QUESTION 249

- (Exam Topic 4)

You cannot install third party plugins using terraform init.

- A. True
- B. False

Answer: B

Explanation:

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

For providers that are published in either the public Terraform Registry or in a third-party provider registry, terraform init will automatically find, download, and install the necessary provider plugins.

NEW QUESTION 254

- (Exam Topic 4)

After executing a terraform apply, you notice that a resource has a tilde (~) next to it. What does this infer?

- A. The resource will be updated in place.
- B. The resource will be created.
- C. Terraform can't determine how to proceed due to a problem with the state file.
- D. The resource will be destroyed and recreated.

Answer: A

Explanation:

The prefix +/- means that Terraform will destroy and recreate the resource, rather than updating it in-place. The prefix ~ means that some attributes and resources can be updated in-place.

```
$ terraform apply
```

```
aws_instance.example: Refreshing state... [id=i-0bbf06244e44211d1] An execution plan has been generated and is shown below.
```

Resource actions are indicated with the following symbols:

```

-/+ destroy and then create replacement Terraform will perform the following actions:
# aws_instance.example must be replaced
-/+ resource "aws_instance" "example" {
~ ami = "ami-2757f631" -> "ami-b374d5a5" # forces replacement
~ arn = "arn:aws:ec2:us-east-1:130490850807:instance/i-0bbf06244e44211d1" -> (known after apply)
~ associate_public_ip_address = true -> (known after apply)
~ availability_zone = "us-east-1c" -> (known after apply)
~ cpu_core_count = 1 -> (known after apply)
~ cpu_threads_per_core = 1 -> (known after apply)
- disable_api_termination = false -> null
- ebs_optimized = false -> null get_password_data = false
+ host_id = (known after apply)
~ id = "i-0bbf06244e44211d1" -> (known after apply)
~ instance_state = "running" -> (known after apply) instance_type = "t2.micro"
~ ipv6_address_count = 0 -> (known after apply)
~ ipv6_addresses = [] -> (known after apply)
+ key_name = (known after apply)
- monitoring = false -> null
+ network_interface_id = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
~ primary_network_interface_id = "eni-0f1ce5bdae258b015" -> (known after apply)
~ private_dns = "ip-172-31-61-141.ec2.internal" -> (known after apply)
~ private_ip = "172.31.61.141" -> (known after apply)
~ public_dns = "ec2-54-166-19-244.compute-1.amazonaws.com" -> (known after apply)
~ public_ip = "54.166.19.244" -> (known after apply)
~ security_groups = [
- "default",
] -> (known after apply) source_dest_check = true
~ subnet_id = "subnet-1facdf35" -> (known after apply)
~ tenancy = "default" -> (known after apply)
~ volume_tags = {} -> (known after apply)
~ vpc_security_group_ids = [
- "sg-5255f429",
] -> (known after apply)
- credit_specification {
- cpu_credits = "standard" -> null
}
+ ebs_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ snapshot_id = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
+ ephemeral_block_device {
+ device_name = (known after apply)
+ no_device = (known after apply)
+ virtual_name = (known after apply)
}
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
~ root_block_device {
~ delete_on_termination = true -> (known after apply)
~ iops = 100 -> (known after apply)
~ volume_id = "vol-0079e485d9e28a8e5" -> (known after apply)
~ volume_size = 8 -> (known after apply)
~ volume_type = "gp2" -> (known after apply)
}
}
}
Plan: 1 to add, 0 to change, 1 to destroy.

```

NEW QUESTION 255

- (Exam Topic 4)

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

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

Answer: C

NEW QUESTION 258

- (Exam Topic 4)

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

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

Answer: ADE

NEW QUESTION 259

- (Exam Topic 4)

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_10G_PATM
- B. TF_LOG
- C. TF_10G_LEVEL
- D. TF.LOG.FUE

Answer: B

Explanation:

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

NEW QUESTION 261

- (Exam Topic 4)

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

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

Answer: ABC

Explanation:

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

NEW QUESTION 262

- (Exam Topic 4)

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:

Wherever possible, avoid using terraform refresh explicitly and instead rely on Terraform's behavior of automatically refreshing existing objects as part of creating a normal plan. Source: <https://www.terraform.io/cli/commands/refresh>

NEW QUESTION 263

- (Exam Topic 4)

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

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

Answer: B

Explanation:

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

NEW QUESTION 264

- (Exam Topic 4)

Given the Terraform configuration below, in which order will the resources be created?

```
* 1. resource "aws_instance" "web_server"
* 2. {
* 3. ami = "ami-b374d5a5"
* 4. instance_type = "t2.micro"
* 5. }
* 6. resource "aws_eip" "web_server_ip"
* 7. {
* 8. vpc = true instance = aws_instance.web_server.id
* 9. }
```

- A. aws_eip will be created first aws_instance will be created second
- B. aws_eip will be created first aws_instance will be created second
- C. Resources will be created simultaneously
- D. aws_instance will be created first aws_eip will be created second

Answer: D

Explanation:

Implicit and Explicit Dependencies

By studying the resource attributes used in interpolation expressions, Terraform can automatically infer when one resource depends on another. In the example above, the reference to `aws_instance.web_server.id` creates an implicit dependency on the `aws_instance` named `web_server`.

Terraform uses this dependency information to determine the correct order in which to create the different resources.

```
# Example of Implicit Dependency resource "aws_instance" "web_server" { ami = "ami-b374d5a5"
instance_type = "t2.micro"
}
resource "aws_eip" "web_server_ip" { vpc = true
instance = aws_instance.web_server.id
}
```

In the example above, Terraform knows that the `aws_instance` must be created before the `aws_eip`. Implicit dependencies via interpolation expressions are the primary way to inform Terraform about these relationships, and should be used whenever possible.

Sometimes there are dependencies between resources that are not visible to Terraform. The `depends_on` argument is accepted by any resource and accepts a list of resources to create explicit dependencies for.

For example, perhaps an application we will run on our EC2 instance expects to use a specific Amazon S3 bucket, but that dependency is configured inside the application code and thus not visible to Terraform. In that case, we can use `depends_on` to explicitly declare the dependency:

```
# Example of Explicit Dependency
# New resource for the S3 bucket our application will use. resource "aws_s3_bucket" "example" {
bucket = "terraform-getting-started-guide" acl = "private"
}
# Change the aws_instance we declared earlier to now include "depends_on" resource "aws_instance" "example" {
ami = "ami-2757f631" instance_type = "t2.micro"
# Tells Terraform that this EC2 instance must be created only after the
# S3 bucket has been created. depends_on = [aws_s3_bucket.example]
}
https://learn.hashicorp.com/terraform/getting-started/dependencies.html
```

NEW QUESTION 265

- (Exam Topic 4)

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

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

Answer: C

Explanation:

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

NEW QUESTION 268

- (Exam Topic 4)

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

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

Answer: A

NEW QUESTION 271

- (Exam Topic 4)

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

- A. False
- B. True

Answer: B

NEW QUESTION 273

- (Exam Topic 4)

Your firm employs a version control system (for example, git) and has requested that you commit all terraform code to it. During the commit, you must be cautious with sensitive information. Which of the following files should be left out of the commit?

- A. main.tf
- B. variables.tf
- C. provisioner.tf
- D. terraform.tfstate

Answer: D

NEW QUESTION 276

- (Exam Topic 4)

What Terraform feature is shown in the example below?

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

Answer: C

NEW QUESTION 277

- (Exam Topic 4)

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

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

Answer: D

Explanation:

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

NEW QUESTION 282

- (Exam Topic 4)

When writing Terraform code, HashiCorp recommends that you use how many spaces between each nesting level?

- A. 1
- B. 2
- C. 4

Answer: C

Explanation:

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.

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:

```
ami = "abc123" instance_type = "t2.micro"
```

When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them.

Use one blank line to separate the arguments from the blocks.

Use empty lines to separate logical groups of arguments within a block.

For blocks that contain both arguments and "meta-arguments" (as defined by the Terraform language semantics), list meta-arguments first and separate them from other arguments with one blank line. Place meta-argument blocks last and separate them from other blocks with one blank line.

```
resource "aws_instance" "example" { count = 2 # meta-argument first
```

```
ami = "abc123" instance_type = "t2.micro" network_interface {
```

```
# ...
```

```
}
```

```
lifecycle { # meta-argument block last create_before_destroy = true
```

```
}
```

```
}
```

Top-level blocks should always be separated from one another by one blank line. Nested blocks should also be separated by blank lines, except when grouping together related blocks of the same type (like multiple provisioner blocks in a resource).

Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example: `root_block_device`, `ebs_block_device` and `ephemeral_block_device` on `aws_instance` form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

NEW QUESTION 284

- (Exam Topic 4)

What does terraform refresh modify?

- A. Your cloud infrastructure
- B. Your state file

- C. Your Terraform plan
- D. Your Terraform configuration

Answer: B

Explanation:

The terraform refresh command reads the current settings from all managed remote objects and updates the Terraform state to match. Source: <https://www.terraform.io/cli/commands/refresh>

NEW QUESTION 287

- (Exam Topic 4)

Most Terraform providers interact with _____.

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

Answer: A

Explanation:

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

NEW QUESTION 289

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your TA-002-P Exam with Our Prep Materials Via below:

<https://www.certleader.com/TA-002-P-dumps.html>