Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

https://www.2passeasy.com/dumps/TA-002-P/

NEW QUESTION 1
- (Exam Topic 1)
When you initialize Terraform, where does it cache modules from the public Terraform Module Registry?

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

Answer: C

Explanation:

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

NEW QUESTION 2
- (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 3
- (Exam Topic 1)
Terraform provisioners can be added to any resource block.

A. True
B. False

Answer: A

Explanation:
https://www.phillipsj.net/posts/introduction-to-terraform-provisioners/
As you continue learning about Terraform, you will start hearing about provisioners. Terraform provisioners can be created on any resource and provide a way to execute actions on local or remote machines.
https://www.terraform.io/language/resources/provisioners/local-exec

NEW QUESTION 4
- (Exam Topic 1)
You need to deploy resources into two different cloud regions in the same Terraform configuration. To do that, you declare multiple provider configurations as follows:

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

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

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

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

Answer: C

Explanation:

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

NEW QUESTION 5
- (Exam Topic 1)
Which argument(s) is (are) required when declaring a Terraform variable?

A. type
B. default
C. description
D. All of the above
E. None of the above

Answer: B

Explanation:
The variable declaration can also include a default argument.
Reference: https://www.terraform.io/docs/language/values/variables.html

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

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

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

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

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

Answer: C

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

NEW QUESTION 10
- (Exam Topic 1)
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:
https://www.terraform.io/cloud-docs/workspaces
"When run locally, Terraform manages each collection of infrastructure with a persistent working directory, which contains a configuration, state data, and variables. Since Terraform CLI uses content from the directory it runs in, you can organize infrastructure resources into meaningful groups by keeping their configurations in separate directories.

NEW QUESTION 10
- (Exam Topic 1)
A terraform apply can not _____ infrastructure.

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

Answer: D

Explanation:
https://www.educative.io/answers/what-is-the-command-to-destroy-infrastructure-in-terraform

NEW QUESTION 15
- (Exam Topic 1)
Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.

A. True
B. False

Answer: B

Explanation:
 https://www.terraform.io/downloads

NEW QUESTION 17
- (Exam Topic 1)
You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (files). You need to enable debug messages to find this out.
Which of the following would achieve this?

A. Set the environment variable TF_LOG=TRACE
B. Set verbose logging for each provider in your Terraform configuration
C. Set the environment variable TF_VAR_log=TRACE
D. Set the environment variable TF_LOG_PATH

Answer: A

Explanation:
Although this will only output to stderr and if you need to review log file you will need to include TF_LOG_PATH=pathtofile
https://www.terraform.io/internals/debugging

NEW QUESTION 22
- (Exam Topic 1)
You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.
What are the two things you must do to achieve this? (Choose two.)

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

Answer: BC

Explanation:
You should create the equivalent configuration first, and then run import to load it on the state file.

NEW QUESTION 26
- (Exam Topic 1)
Only the user that generated a plan may apply it.

A. True
B. False

Answer: B

NEW QUESTION 31
- (Exam Topic 1)
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 local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource."
https://www.terraform.io/language/resources/provisioners/local-exec
"The remote-exec provisioner invokes a script on a remote resource after it is created." https://www.terraform.io/language/resources/provisioners/remote-exec

NEW QUESTION 36
- (Exam Topic 1)
You just scaled your VM infrastructure and realized you set the count variable to the wrong value. You correct the value and save your change.
What do you do next to make your infrastructure match your configuration?

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

Answer: A

NEW QUESTION 41
- (Exam Topic 1)
How would you reference the "name" value of the second instance of this fictitious resource?

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

A. element(aws_instance.web, 2)
B. aws_instance.web[1].name

C. aws_instance.web[1]
D. aws_instance.web[2].name
E. aws_instance.web.*.name

Answer: B

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

NEW QUESTION 46
- (Exam Topic 1)
FILL BLANK
Which flag would you add to terraform plan to save the execution plan to a file?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

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

NEW QUESTION 47
- (Exam Topic 1)
Terraform variables and outputs that set the "description" argument will store that description in the state file.

A. True
B. False

Answer: B

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

NEW QUESTION 49
- (Exam Topic 1)
A Terraform local value can reference other Terraform local values.

A. True
B. False

Answer: A

Explanation:
"The expressions in local values are not limited to literal constants; they can also reference other values in the module in order to transform or combine them, including variables, resource attributes, or other local values:" https://www.terraform.io/language/values/locals#declaring-a-local-value

NEW QUESTION 52
- (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 53
- (Exam Topic 1)
What is the name assigned by Terraform to reference this resource?

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

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

Answer: B

NEW QUESTION 55
- (Exam Topic 1)
What command does Terraform require the first time you run it within a configuration directory?

A. terraform import
B. terraform init
C. terraform plan
D. terraform workspace

Answer: B

Explanation:
terraform init command is used to initialize a working directory containing Terraform configuration files. Reference:
https://www.terraform.io/docs/cli/commands/init.html

NEW QUESTION 56
- (Exam Topic 1)
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:
https://www.terraform.io/cdktf/concepts/remote-backends https://docs.gitlab.com/ee/user/infrastructure/iac/terraform_state.html

NEW QUESTION 61
- (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 63
- (Exam Topic 1)
You have multiple team members collaborating on infrastructure as code (IaC) using Terraform, and want to apply formatting standards for readability.
How can you format Terraform HCL (HashiCorp Configuration Language) code according to standard Terraform style convention?

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

Answer: A

Explanation:
https://www.terraform.io/cli/commands/fmt

NEW QUESTION 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 68
- (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 69
- (Exam Topic 2)
Terraform has detailed logs which can be enabled by setting the _____ environmental variable.

A. TF_TRACE
B. TF_DEBUG
C. TF_LOG
D. TF_INFO

Answer: C

Explanation:
Terraform has detailed logs that 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. https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION 71
- (Exam Topic 2)
John wants to use two different regions to deploy two different EC2 instances. He has specified two provider blocks in his providers.tf file.
provider "aws" { region = "us-east-1" } provider "aws" { region = "us-west-2" }
When he run terraform plan he encountered an error. How to fix this?

A. Use another provider version
B. Use alias for region = "us-west-2"
C. Use default keyword with region = "us-east-1"
D. It can not be fixed

Answer: B

NEW QUESTION 75
- (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 78
- (Exam Topic 2)
Which one of the following will run echo 0 and echo 1 on a newly created host?

A. provisioner "local-exec" { command = "echo 0" command = "echo 1"}
B. provisioner "remote-exec" { inline = [echo 0,echo 1]}
C. provisioner "remote-exec" {command = "${echo 0}" command = "${echo 1}"}
D. provisioner "remote-exec" { inline = ["echo 0","echo 1"]}

Answer: D

Explanation:
remote-exec Provisioner Example usage
resource "aws_instance" "web" {
# ...
provisioner "remote-exec" { inline = [
"puppet apply",
"consul join ${aws_instance.web.private_ip}",
]
}
}

NEW QUESTION 83
- (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 84
- (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 85
- (Exam Topic 2)
Which Terraform command will force a marked resource to be destroyed and recreated on the next apply?

A. terraform fmt
B. terraform destroy
C. terraform taint
D. terraform refresh

Answer: C

Explanation:
The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.
Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.
Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.
https://www.terraform.io/docs/commands/taint.html

NEW QUESTION 90
- (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 92
- (Exam Topic 2)
Which of the following Terraform files should be ignored by Git when committing code to a repo? (select Three)

A. Files named exactly terraform.tfvars or terraform.tfvars.json.
B. Any files with names ending in .auto.tfvars or .auto.tfvars.json.
C. input.tf
D. terraform.tfstate
E. output.tf

Answer: ABD

Explanation:
The .gitignore file should be configured to ignore Terraform files that either contain sensitive data or are not required to save.
Terraform state (terraform.tfstate) 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.
The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others.

NEW QUESTION 93
- (Exam Topic 2)
terraform refresh will update 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. If the state is changed, this may cause changes to occur during the next plan or apply.

NEW QUESTION 97
- (Exam Topic 2)
What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

A. Local backends
B. Providers
C. Remote backends
D. Workspaces

Answer: D

Explanation:
Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure.
Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION 98
- (Exam Topic 2)
You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?

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

Answer: B

Explanation:

Command: force-unlock

Manually unlock the state for the defined configuration.

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

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

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

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

NEW QUESTION 103
- (Exam Topic 2)
How does Terraform handle working with so many providers?

A. Terraform ships with all of the plugins embedded in the Terraform binary.
B. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in the configuration's working directory.
C. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in a shared, system-wide plugins directory.
D. Terraform allows you to select the providers you want to support during the Terraform installation process.

Answer: B

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

NEW QUESTION 104
- (Exam Topic 2)
When using remote state, state is only ever held in memory when used by Terraform.

A. False
B. True

Answer: B

NEW QUESTION 106
- (Exam Topic 2)
Which of the following command can be used to view the specified version constraints for all providers used in the current configuration.

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

Answer: A

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

NEW QUESTION 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 111
- (Exam Topic 2)
What is the purpose of using the local-exec provisioner? (Select Two)

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

Answer: AC

Explanation:
The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource.

Note that even though the resource will be fully created when the provisioner is run, there is no guarantee that it will be in an operable state - for example system services such as sshd may not be started yet on compute resources.
Example usage
resource "aws_instance" "web" {
# ...
provisioner "local-exec" {
command = "echo ${aws_instance.web.private_ip} >> private_ips.txt"
}
}
Note: Provisioners should only be used as a last resort. For most common situations there are better alternatives.
https://www.terraform.io/docs/provisioners/local-exec.html

NEW QUESTION 112
- (Exam Topic 2)
What does terrafom plan do ?

A. Create an execution plan by evaluating the difference between configuration file and state file.
B. Performs a refresh, unless explicitly disabled, and then apply the changes that are necessary to achieve the desired state specified in the configuration files.
C. Create an execution plan by evaluating the difference between configuration file and actual infrastructure.
D. Checks whether the execution plan for a set of changes matches your expectations by making changes to real resources or to the state.

Answer: A

NEW QUESTION 114
- (Exam Topic 2)
Terraform must track metadata such as resource dependencies. Where is this data stored?

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

Answer: C

Explanation:
Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.
To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state. Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.
https://www.terraform.io/docs/state/purpose.html#metadata

NEW QUESTION 115
- (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 120
- (Exam Topic 3)
A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the module:
module "infranet" { source = "s3::https://s3-us-west-2.amazonaws.com/infrabucket/infra_module.zip"} What is the best way to ensure that you are not using the latest version of the module?

A. Add a module version constraint in your configuration's backend block and specify a previous version.
B. Add a version key to the module configuration and specify a previous version.
C. Delete the latest version of the module in S3 to rollback to the previous version.
D. Add a version property to the module in Terraform's state file and specify a previous version.

Answer: C

Explanation:
Version constraints are supported only for modules installed from a module registry, such as the 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.
Only Terraform Registries support module versioning by using the version key, one cannot configure a previous version of the module in the configuration. Deleting the latest version of the module in S3 is the only option of the available options that ensures you won't use the latest version. You could also modify the source URL to specify a versionId URL parameter for a previous version.
https://www.terraform.io/docs/configuration/modules.html#source

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 125
- (Exam Topic 3)
Which of the following challenges would Terraform be a candidate for solving? (Select THREE)

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

Answer: ABD

NEW QUESTION 126
- (Exam Topic 3)
Which of the below options is a valid interpolation syntax for retrieving a data source?

A. ${google_storage_bucket.backend}
B. ${azurerm_resource_group.test.data}
C. ${aws_instance.web.id.data}
D. ${data.google_dns_keys.foo_dns_keys.key_signing_keys[0].ds_record}

Answer: D

Explanation:
Data source attributes are interpolated with the general syntax data.TYPE.NAME.ATTRIBUTE. The interpolation for a resource is the same but without the data.
prefix (TYPE.NAME.ATTRIBUTE).
https://www.terraform.io/docs/configuration-0-11/interpolation.html#attributes-of-a-data-source

NEW QUESTION 130
- (Exam Topic 3)
Once a resource is marked as tainted, the next plan will show that the resource will be _____ and _____ and the next apply will implement this change.

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

Answer: D

NEW QUESTION 135
- (Exam Topic 3)
After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?

A. ERROR
B. INFO
C. WARN
D. TRACE
E. DEBUG

Answer: D

Explanation:
Terraform has detailed logs that 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.
Examples:
export TF_LOG=DEBUG export TF_LOG=TRACE

NEW QUESTION 140
- (Exam Topic 3)
Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.

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

Answer: BC

NEW QUESTION 144
- (Exam Topic 3)
Which of the following variable definition files will terraform load automatically?

A. terraform.tfvar
B. Any files with names ending in .auto.tfvars.json
C. terraform.tfvars
D. terraform.tfvars.json

Answer: BCD

Explanation:
Terraform also automatically loads a number of variable definitions files if they are present: Files named exactly terraform.tfvars or terraform.tfvars.json.
Any files with names ending in .auto.tfvars or .auto.tfvars.json. https://www.terraform.io/docs/configuration/variables.html
https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files

NEW QUESTION 146
- (Exam Topic 3)
Complete the following sentence:
For local state, the workspaces are stored directly in a _____.

A. a file called terraform.tfstate.backup
B. directory called terraform.workspaces.tfstate
C. a file called terraform.tfstate
D. directory called terraform.tfstate.d

Answer: D

Explanation:
For local state, Terraform stores the workspace states in a directory called terraform.tfstate.d. https://www.terraform.io/docs/state/workspaces.html#workspace-internals

NEW QUESTION 147
- (Exam Topic 3)
Terraform Enterprise currently supports running under which the following operating systems?

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

Answer: ABCDEF

Explanation:
Terraform Enterprise runs on Linux instances, and you must prepare a running Linux instance for Terraform Enterprise before running the installer. You will start and manage this instance like any other server.
Terraform Enterprise currently supports running under the following operating systems: Standalone deployment:
Debian 7.7+
Ubuntu 14.04.5 / 16.04 / 18.04
Red Hat Enterprise Linux 7.4 - 7.8 CentOS 6.x / 7.4 - 7.8
Amazon Linux 2014.03 / 2014.09 / 2015.03 / 2015.09 / 2016.03 / 2016.09 / 2017.03 / 2017.09 / 2018.03 / 2.0
Oracle Linux 7.4 - 7.8 https://www.terraform.io/docs/enterprise/before-installing/index.html

NEW QUESTION 148
- (Exam Topic 3)
You cannot publish your own modules on the Terraform Registry.

A. False
B. True

Answer: A

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

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

NEW QUESTION 152
- (Exam Topic 3)
Hanah is writing a terraform configuration with nested modules, there are multiple places where she has to use the same conditional expression but she wants to avoid repeating the same values or expressions multiple times in the configuration,. What is a better approach to dealing with this?

A. Expressions
B. Local Values
C. Variables
D. Functions

Answer: B

Explanation:
https://www.terraform.io/docs/configuration/locals.html

NEW QUESTION 157
- (Exam Topic 3)
When using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects. Which of the below option is a recommended solution for this?

A. Remote State
B. Module
C. Use the cached state and treat this as the record of truth.
D. Workspace

Answer: A

Explanation:
https://www.terraform.io/docs/state/remote.html

NEW QUESTION 160
- (Exam Topic 4)
Valarie has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code. Valarie wants to hide the output value in the CLI after terraform apply that's why she has used sensitive parameter.
* 1. output "db_password" {
* 2. value = local.db_password
* 3. sensitive = true
* 4. }
Since sensitive is set to true, will the value associated with db password be available in plain-text in the state file for everyone to read?

A. Yes
B. No

Answer: A

Explanation:
Outputs can be marked as containing sensitive material by setting the sensitive attribute to true, like this: output "sensitive" {
sensitive = true value = VALUE
}
When outputs are displayed on-screen following a terraform apply or terraform refresh, sensitive outputs are redacted, with <sensitive> displayed in place of their value.
Limitations of Sensitive Outputs
The values of sensitive outputs are still stored in the Terraform state, and available using the terraform output command, so cannot be relied on as a sole means of protecting values.
Sensitivity is not tracked internally, so if the output is interpolated in another module into a resource, the value will be displayed.

NEW QUESTION 161
- (Exam Topic 4)
What is the purpose of a Terraform workspace in either open source of 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: B

NEW QUESTION 162
- (Exam Topic 4)
How would you reference the attribute "name* of this fictitious resource in HCL?

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

A. resource.kubrnetes_namespace>example.name
B. kubernetes_namespace.test.name
C. kubernetes_namespace.example,name
D. data kubernetes_namespace.name
E. None of the above

Answer: C

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

NEW QUESTION 165
- (Exam Topic 4)
Which of the following actions are performed during a terraform init?

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

Answer: ABD

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

NEW QUESTION 169
- (Exam Topic 4)
You have configured an Auto Scaling group in AWS to automatically scale the number of instances behind a load balancer based on the instances CPU utilization. The instances are configured using a Launch Configuration. You have observed that the Auto Scaling group doesn't successfully scale when you apply changes that require replacing the Launch Configuration. Why is this happening?

A. You need to configure an explicit dependency for the Auto Scaling group using the depends_on meta-parameter.
B. You need to configure an explicit dependency for the Launch Configuration using the depends_on meta-parameter.
C. You need to configure the Auto Scaling group's create_before_destroy meta-parameter.
D. You need to configure the Launch Configuration's create_before_destroy meta-parameter.

Answer: D

Explanation:
https://www.terraform.io/docs/providers/aws/r/launch_configuration.html#using-withautoscaling-groups

NEW QUESTION 171
- (Exam Topic 4)
Open source Terraform can only import publicly-accessible and open-source modules.

A. True
B. False

Answer: B

Explanation:
Terraform can load modules from a public or private registry. This makes it possible to publish modules for others to use, and to use modules that others have published. Also, members of your organization might produce modules specifically crafted for your own infrastructure needs. Terraform Cloud and Terraform Enterprise both include a private module registry for sharing modules internally within your organization. Source: https://www.terraform.io/language/modules

NEW QUESTION 174
- (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 175
- (Exam Topic 4)
John is writing a module and within the module, there are multiple places where he has to use the same
conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration,. What is a better approach to dealing with this?

A. Local Values
B. Expressions
C. Functions
D. Variables

Answer: A

Explanation:
A local value assigns a name to an expression, allowing it to be used multiple times within a module without repeating it.
https://www.terraform.io/docs/configuration/locals.html

NEW QUESTION 179
- (Exam Topic 4)
Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)

A. Explicit use of a provider block in configuration, optionally including a version constraint.
B. Use of any resource belonging to a particular provider in a resource or data block in configuration.
C. Existence of any resource instance belonging to a particular provider in the current state.
D. Existence of any provider plugins found locally in the working directory.

Answer: ABC

Explanation:
The existence of a provider plugin found locally in the working directory does not itself create a provider dependency. The plugin can exist without any reference to it in the terraform configuration. https://www.terraform.io/docs/commands/providers.html

NEW QUESTION 181
- (Exam Topic 4)
In the example below, where is the value of the DNS record's IP address originating from?
* 1. resource "aws_route53_record" "www"
* 2. {
* 3. zone_id = aws_route53_zone.primary.zone_id
* 4. name = "www.example.com"
* 5. type = "A"
* 6. ttl = "300"
* 7. records = [module.web_server.instance_ip_address] 8. }

A. The regular expression named module.web_server
B. The output of a module named web_server
C. By querying the AWS EC2 API to retrieve the IP address
D. Value of the web_server parameter from the variables.tf file

Answer: B

Explanation:
In a parent module, outputs of child modules are available in expressions as module.<MODULE NAME>.<OUTPUT NAME>.
For example, if a child module named web_server declared an output named instance_ip_address, you could access that value as
module.web_server.instance_ip_address.

NEW QUESTION 183
- (Exam Topic 4)
From the code below, identify the implicit dependency:

A. The EIP with an id of ami-2757f631
B. The AMI used for the EC2 instance
C. The EC2 instance labeled web_server
D. The S3 bucket labeled company_data

Answer: C

NEW QUESTION 185
- (Exam Topic 4)
Your configuration file has been locked accidentally. What of the following command would you use to unlock?

A. terraform filename-unlock
B. delete the file and create a new state file
C. terraform force-unlock
D. state.tf-unlock

Answer: C

NEW QUESTION 188
- (Exam Topic 4)
A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?

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

Answer: B

Explanation:
The terraform workspace select command is used to choose a different workspace to use for further operations.
https://www.terraform.io/docs/commands/workspace/select.html

NEW QUESTION 193
- (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 195
- (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 199
- (Exam Topic 4)
Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements. (select three)

A. The module must be PCI/HIPPA compliant.
B. Module repositories must use this three-part name format, terraform-- .
C. The registry uses tags to identify module versions.
D. Release tag names must be for the format x.y.z, and can optionally be prefixed with a v .
E. The module must be on GitHub and must be a public repo.

Answer: CDE

Explanation:
https://www.terraform.io/docs/registry/modules/publish.html#requirements

NEW QUESTION 202
- (Exam Topic 4)
What is the result of the following terraform function call?

A. hello
B. what?
C. goodbye

Answer: B

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

NEW QUESTION 206
- (Exam Topic 4)
True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

A. False
B. True

Answer: B

Explanation:
The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

NEW QUESTION 207
- (Exam Topic 4)
colleagues is new toTerraform and wants to add a new workspace named new-hire. What command he should execute from the following?

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

Answer: B

NEW QUESTION 212
- (Exam Topic 4)
How would you reference the Volume IDs associated with the ebs_block_device blocks in this configuration?

```
resource "aws_instance" "example" {
  ami = "ami-abc123"
  instance_type - "t2.micro"

ebs_block_device {
  device_name = "sda2"
  volume_size = 16
}

ebs_block_device {
  device_name = "sda3"
  volume_size = 20
  }
}
```

A. aws_instance.example.ebs_block_device.[*].volume_id
B. aws_instance.example.ebs_block_device.volume_id
C. aws_instance.example.ebs_block_device[sda2,sda3].volume_id
D. aws_instance.example.ebs_block_device.*.volume_id

Answer: A

Explanation:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/device_naming.html

NEW QUESTION 216
- (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 218
- (Exam Topic 4)
Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?

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

Answer: A

Explanation:
The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.
https://www.terraform.io/docs/commands/console.html

NEW QUESTION 221
- (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 224
- (Exam Topic 4)
Your company has a lot of workloads in AWS , and Azure that were respectively created using CloudFormation , and AzureRM Templates. However , now your CIO has decided to use Terraform for all new projects , and has asked you to check how to integrate the existing environment with terraform code. What should be your next plan of action?

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

Answer: B

NEW QUESTION 227
- (Exam Topic 4)
Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?

A. False
B. True

Answer: B

Explanation:
For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.
For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the -refresh=false flag as well as the -target flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.
https://www.terraform.io/docs/state/purpose.html

NEW QUESTION 228
- (Exam Topic 4)
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:
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.
https://www.terraform.io/language/modules/syntax

NEW QUESTION 230
- (Exam Topic 4)
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

NEW QUESTION 231
- (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 233
- (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 234
- (Exam Topic 4)
In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to use what?

A. Input Parameters
B. Module
C. Regular Expressions
D. Output Value

Answer: A

Explanation:
Input variables serve as parameters for a Terraform module, allowing aspects of the module to be customized without altering the module's own source code, and allowing modules to be shared between different configurations.
https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION 236
- (Exam Topic 4)
When does Sentinel enforce policy logic during a Terraform Enterprise run?

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

Answer: C

Explanation:
"Enforcing policy checks on runs - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is

executed."

NEW QUESTION 239
- (Exam Topic 4)
Jack is a newbieto Terraform and wants to enable detailed logging to find all the details. Which environment variable does he need to set?

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

Answer: B

NEW QUESTION 244
- (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 246
- (Exam Topic 4)
Select the feature below that best completes the sentence:
The following list represents the different types of _____ available in Terraform.
* 1. max
* 2. min
* 3. join
* 4. replace
* 5. list
* 6. length
* 7. range

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

Answer: D

Explanation:
The Terraform language includes a number of built-in functions that you can call from within expressions to transform and combine values. The Terraform language does not support user-defined functions, and only the functions built into the language are available for use.
https://www.terraform.io/docs/configuration/functions.html

NEW QUESTION 250
- (Exam Topic 4)
What kind of configuration block will create an infrastructure object with settings specified in the block?

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

Answer: C

NEW QUESTION 253
- (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)
When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.
For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.

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

Answer: C

Explanation:
Output values are like the return values of a Terraform module, and have several uses:
* A child module can use outputs to expose a subset of its resource attributes to a parent module.
* A root module can use outputs to print certain values in the CLI output after running terraform apply.
* When using remote state, root module outputs can be accessed by other configurations via a terraform_remote_state data source.
https://www.terraform.io/docs/configuration/outputs.html

NEW QUESTION 260
- (Exam Topic 4)
Suppose terraformcode is taking up some values which are not defined inside the code files. In which of the following options issue might have occurred?

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

Answer: D

NEW QUESTION 264
- (Exam Topic 4)
What does terraform refresh modify?

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

Answer: C

NEW QUESTION 269
- (Exam Topic 4)
Which are examples of infrastructure as code? (Choose two.)

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

Answer: CD

NEW QUESTION 270
- (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 271
- (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 274
- (Exam Topic 4)

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

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

Answer: D

Explanation:
Data sources allow data to be fetched or computed for use elsewhere in Terraform configuration. Use of data sources allows a Terraform configuration to build on information defined outside of Terraform, or defined by another separate Terraform configuration.

NEW QUESTION 276
- (Exam Topic 4)
You want to define multiple data disks as nested blocks inside the resource block for a virtual machine. What Terraform feature would help you define the blocks using the values in a variable?

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

Answer: B

NEW QUESTION 281
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

Answer: A

NEW QUESTION 282
- (Exam Topic 4)
Which of the following does terraform apply change after you approve the execution plan? Choose two correct answers.

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

Answer: CD

NEW QUESTION 284
- (Exam Topic 4)
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. Source: https://www.terraform.io/language/state/locking

NEW QUESTION 287
- (Exam Topic 4)
A variable az has the following default value. What will be the datatype of the variable? az=["us-west-1a","us-east-1a"]

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

Answer: B

NEW QUESTION 289
- (Exam Topic 4)
What does Terraform use .terraform.lock.hcl file for?

A. Tracking provider dependencies Most Voted
B. There is no such file
C. Preventing Terraform runs from occurring
D. Storing references to workspaces which are locked

Answer: A

Explanation:
https://www.terraform.io/language/files/dependency-lock
"hcl , and this name is intended to signify that it is a lock file for various items that Terraform caches in the . terraform subdirectory of your working directory. Terraform automatically creates or updates the dependency lock file each time you run the terraform init command."

NEW QUESTION 293
- (Exam Topic 4)
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

NEW QUESTION 295
- (Exam Topic 4)
Which Terraform collection type should you use to store key/value pairs?

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

Answer: D

Explanation:
Maps/objects are represented by a pair of curly braces containing a series of <KEY> = <VALUE> pairs Source:
https://www.terraform.io/language/expressions/types

NEW QUESTION 299
- (Exam Topic 4)
You are using a networking module in your Terraform configuration with the name label my_network. In your main configuration you have the following code:

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

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

```
Error: Reference to undeclared output value

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

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

A. Define the attribute vnet_id as a variable in the networking module
B. Change the referenced value to module.my_network.outputs.vnet_id
C. Define the attribute vnet_id as an output in the networking module
D. Change the referenced value to my_network.outputs.vnet_id

Answer: C

Explanation:
In a parent module, outputs of child modules are available in expressions as module.<MODULE NAME>.<OUTPUT NAME>. For example, if a child module named web_server declared an output named instance_ip_addr, you could access that value as module.web_server.instance_ip_addr.

NEW QUESTION 302
- (Exam Topic 4)
Select all features which are exclusive to Terraform Enterprise. (Select Three)

A. Sentinel
B. Cost Estimation
C. Audit Logs
D. Clustering
E. SAML/SSO

Answer: CDE

Explanation:
Sentinel and Cost Estimation are also available in Terraform Cloud https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 307
- (Exam Topic 4)
As a member of the operations team, you need to run a script on a virtual machine created by Terraform. Which provisioned is best to use in your Terraform code?

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

Answer: D

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

NEW QUESTION 309
- (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 311
- (Exam Topic 4)
terraform apply is failing with the following error. What next step should you take to determine the root cause of the problem?
Error loading state: AccessDenied: Access Denied status code: 403, request id: 288766CE5CCA24A0, host id: FOOBAR

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

Answer: A

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

NEW QUESTION 314
- (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 318
- (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 319
- (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 320
- (Exam Topic 4)
You have been working in a Cloud provider account that is shared with other team members. You previously used Terraform to create a load balancer that is listening on port 80. After some application changes, you updated the Terraform code to change the port to 443.
You run terraform plan and see that the execution plan shows the port changing from 80 to 443 like you intended, and step away to grab some coffee.
In the meantime, another team member manually changes the load balancer port to 443 through the Cloud provider console before you get back to your desk.
What will happen when you terraform apply upon returning to your desk?

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

Answer: A

NEW QUESTION 323
......

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual TA-002-P Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the TA-002-P Product From:

https://www.2passeasy.com/dumps/TA-002-P/

Money Back Guarantee

TA-002-P Practice Exam Features:

* TA-002-P Questions and Answers Updated Frequently

* TA-002-P Practice Questions Verified by Expert Senior Certified Staff

* TA-002-P Most Realistic Questions that Guarantee you a Pass on Your FirstTry

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