HashiCorp

Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

NEW QUESTION 1
- (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 2
- (Exam Topic 1)
When you initialize Terraform, where does it cache modules from the public Terraform Module Registry?

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

Answer: C

Explanation:

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

NEW QUESTION 3
- (Exam Topic 1)
Which option can not be used to keep secrets out of Terraform configuration files?

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

Answer: A

Explanation:

Reference: https://secrethub.io/blog/secret-management-for-terraform/

NEW QUESTION 4
- (Exam Topic 1)
You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you delete the VM using the cloud provider console, and run terraform apply again without changing any Terraform code?

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

Answer: D

NEW QUESTION 5
- (Exam Topic 1)
Which of the following is not a valid string function in Terraform?

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

Answer: C

Explanation:

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

NEW QUESTION 6

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

A. True
B. False

Answer: A

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

NEW QUESTION 7
- (Exam Topic 1)
terraform validate validates the syntax of Terraform files.

A. True
B. False

Answer: A

Explanation:
https://www.terraform.io/cli/commands/validate
The terraform validate command validates the syntax and arguments of the Terraform configuration files. Reference:
https://www.terraform.io/docs/cli/code/index.html

NEW QUESTION 8
- (Exam Topic 1)
You need to deploy resources into two different cloud regions in the same Terraform configuration. To do that, you declare multiple provider configurations as follows:

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

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

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

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

Answer: C

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

NEW QUESTION 9
- (Exam Topic 1)
What is the name assigned by Terraform to reference this resource?

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

A. dev
B. azurerm_resource_group
C. azurerm
D. test

Answer: A

NEW QUESTION 10
- (Exam Topic 1)
Terraform can only manage resource dependencies if you set them explicitly with the depends_on argument.

A. True
B. False

Answer: A

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

NEW QUESTION 10
- (Exam Topic 1)
Terraform providers are always installed from the Internet.

A. True
B. False

Answer: B

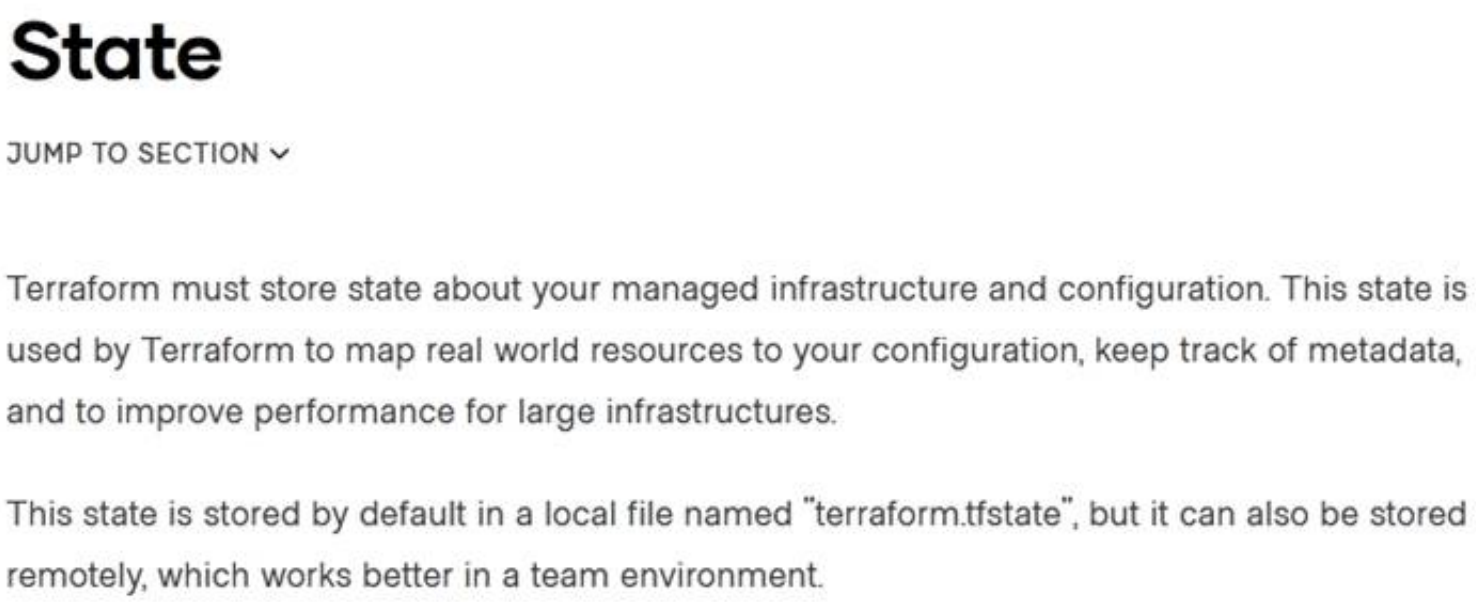
Explanation:
Terraform configurations must declare which providers they require, so that Terraform can install and use them.
Reference: https://www.terraform.io/docs/language/providers/configuration.html

NEW QUESTION 13
- (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



NEW QUESTION 14
- (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 16
- (Exam Topic 1)
How can terraform plan aid in the development process?

A. Validates your expectations against the execution plan without permanently modifying state
B. Initializes your working directory containing your Terraform configuration files
C. Formats your Terraform configuration files

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

Answer: A

Explanation:
"The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. By default, when Terraform creates a plan it:
Reads the current state of any already-existing remote objects to make sure that the Terraform state is up-to-date.
Compares the current configuration to the prior state and noting any differences.
Proposes a set of change actions that should, if applied, make the remote objects match the configuration."
"The plan command alone will not actually carry out the proposed changes, and so you can use this command to check whether the proposed changes match what you expected before you apply the changes or share your changes with your team for broader review.
If Terraform detects that no changes are needed to resource instances or to root module output values, terraform plan will report that no actions need to be taken."
https://www.terraform.io/cli/commands/plan

NEW QUESTION 21
- (Exam Topic 1)
Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)

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

Answer: BD

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

NEW QUESTION 22
- (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 25
- (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 26
- (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 27
- (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 29
- (Exam Topic 1)
You're building a CI/CD (continuous integration/ continuous delivery) pipeline and need to inject sensitive variables into your Terraform run.
How can you do this safely?

A. Pass variables to Terraform with a –var flag
B. Copy the sensitive variables into your Terraform code
C. Store the sensitive variables in a secure_vars.tf file
D. Store the sensitive variables as plain text in a source code repository

Answer: A

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

NEW QUESTION 31
- (Exam Topic 1)
Terraform can run on Windows or Linux, but it requires a Server version of the Windows operating system.

A. True
B. False

Answer: B

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

NEW QUESTION 33
- (Exam Topic 1)
You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (files). You need to enable debug messages to find this out.
Which of the following would achieve this?

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

Answer: A

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

NEW QUESTION 36
- (Exam Topic 1)
Which statement describes a goal of infrastructure as code?

A. An abstraction from vendor specific APIs
B. Write once, run anywhere

C. A pipeline process to test and deliver software
D. The programmatic configuration of resources

Answer: D

Explanation:
The purpose of infrastructure as code is to enable developers or operations teams to automatically manage, monitor and provision resources, rather than manually configure discrete hardware devices and operating systems. Infrastructure as code is sometimes referred to as programmable or software-defined infrastructure.

NEW QUESTION 41
- (Exam Topic 1)
Which of the following is allowed as a Terraform variable name?

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

Answer: B

Explanation:
"The name of a variable can be any valid identifier except the following: source, version, providers, count, for_each, lifecycle, depends_on, locals."
https://www.terraform.io/language/values/variables

NEW QUESTION 46
- (Exam Topic 1)
Which of these options is the most secure place to store secrets foe connecting to a Terraform remote backend?

A. Defined in Environment variables
B. Inside the backend block within the Terraform configuration
C. Defined in a connection configuration outside of Terraform
D. None of above

Answer: A

Explanation:
https://www.terraform.io/language/settings/backends/configuration#credentials-and-sensitive-data Warning: We recommend using environment variables to supply credentials and other sensitive data. If you
use -backend-config or hardcode these values directly in your configuration, Terraform will include these values in both the .terraform subdirectory and in plan files. This can leak sensitive credentials.

NEW QUESTION 47
- (Exam Topic 1)
One remote backend configuration always maps to a single remote workspace.

A. True
B. False

Answer: B

Explanation:
The remote backend can work with either a single remote Terraform Cloud workspace, or with multiple similarly-named remote workspaces (like networking-dev and networking-prod). The workspaces block of the backend configuration determines which mode it uses: To use a single remote Terraform Cloud workspace, set workspaces.name to the remote workspace's full name (like networking-prod). To use multiple remote workspaces, set workspaces.prefix to a prefix used in all of the desired remote workspace names. For example, set prefix = "networking-" to use Terraform cloud workspaces with names like networking-dev and networking-prod. This is helpful when mapping multiple Terraform CLI workspaces used in a single Terraform configuration to multiple Terraform Cloud workspaces.

NEW QUESTION 51
- (Exam Topic 1)
Which of the following is not a key principle of infrastructure as code?

A. Versioned infrastructure
B. Golden images
C. Idempotence
D. Self-describing infrastructure

Answer: B

Explanation:
Reference: https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code#:~:text=Idempotence%20is%20a%20principle%20of,of%20the%20environment's%20starting%20state.

NEW QUESTION 53
- (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 56
- (Exam Topic 1)
What features does the hosted service Terraform Cloud provide? (Choose two.)

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

Answer: CD

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

NEW QUESTION 60
- (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 61
- (Exam Topic 1)
What type of block is used to construct a collection of nested configuration blocks?

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

Answer: D

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

NEW QUESTION 65
- (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 69
- (Exam Topic 1)
You have declared an input variable called environment in your parent module. What must you do to pass the value to a child module in the configuration?

A. Add node_count = var.node_count
B. Declare the variable in a terraform.tfvars file
C. Declare a node_count input variable for child module
D. Nothing, child modules inherit variables of parent module

Answer: C

Explanation:
"That module may call other modules and connect them together by passing output values from one to input values of another."
https://www.terraform.io/language/modules/develop

NEW QUESTION 74
- (Exam Topic 1)
If a module declares a variable with a default, that variable must also be defined within the module.

A. True
B. False

Answer: B

NEW QUESTION 77
- (Exam Topic 1)
When does terraform apply reflect changes in the cloud environment?

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

Answer: B

NEW QUESTION 79
- (Exam Topic 1)
HashiCorp Configuration Language (HCL) supports user-defined functions.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/language/functions
The Terraform language does not support user-defined functions, and so only the functions built into the language are available for use

NEW QUESTION 80
- (Exam Topic 1)
When using a module block to reference a module stored on the public Terraform Module Registry such as:

```
module "consul" {
  source = "hashicorp/consul/aws"
}
```

How do you specify version 1.0.0?

A. Modules stored on the public Terraform Module Registry do not support versioning
B. Append ?ref=v1.0.0 argument to the source path
C. Add version = "1.0.0" attribute to module block
D. Nothing – modules stored on the public Terraform Module Registry always default to version 1.0.0

Answer: C

Explanation:
Version
When using modules installed from a module registry, we recommend explicitly constraining the acceptable version numbers to avoid unexpected or unwanted changes.
Use the version argument in the module block to specify versions: module "consul" {
source = "hashicorp/consul/aws" version = "0.0.5"
servers = 3
}
Reference: https://www.terraform.io/docs/language/modules/sources.html

NEW QUESTION 82
- (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 86
- (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 90
- (Exam Topic 1)
Setting the TF_LOG environment variable to DEBUG causes debug messages to be logged into syslog.

A. True
B. False

Answer: B

Explanation:
TF_LOG_PATH IS NOT REQUIRED, in the docs, they do not mention HAVE TO SET TF_LOG_PATH, it is optional, therefore without TF_LOG_PATH will cause detailed logs to appear on stderr.
https://www.computerhope.com/jargon/s/stderr.htm#:~:text=Stderr%2C%20also%20known%20as%20standard,

NEW QUESTION 93
- (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 97
- (Exam Topic 1)
Which of the following is not a valid Terraform collection type?

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

Answer: C

Explanation:
https://www.terraform.io/language/expressions/type-constraints#collection-types

NEW QUESTION 99
- (Exam Topic 1)
You have deployed a new webapp with a public IP address on a clod provider. However, you did not create any outputs for your code.
What is the best method to quickly find the IP address of the resource you deployed?

A. Run terraform output ip_address to view the result
B. In a new folder, use the terraform_remote_state data source to load in the state file, then write an output for each resource that you find the state file
C. Run terraform state list to find the name of the resource, then terraform state show to find the attributes including public IP address
D. Run terraform destroy then terraform apply and look for the IP address in stdout

Answer: C

Explanation:
https://www.terraform.io/cli/commands/state/show

NEW QUESTION 100
- (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 104
- (Exam Topic 1)
Which of the following is not true of Terraform providers?

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

Answer: E

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

NEW QUESTION 107
- (Exam Topic 1)
How can you trigger a run in a Terraform Cloud workspace that is connected to a Version Control System (VCS) repository?

A. Only Terraform Cloud organization owners can set workspace variables on VCS connected workspaces
B. Commit a change to the VCS working directory and branch that the Terraform Cloud workspace is connected to
C. Only members of a VCS organization can open a pull request against repositories that are connected to Terraform Cloud workspaces
D. Only Terraform Cloud organization owners can approve plans in VCS connected workspaces

Answer: B

Explanation:
"In a workspace linked to a VCS repository, runs start automatically when you merge or commit changes to version control.
A workspace is linked to one branch of a VCS repository and ignores changes to other branches. You can specify which files and directories within your repository trigger runs. "
https://www.terraform.io/cloud-docs/run/ui#automatically-starting-runs

NEW QUESTION 111
- (Exam Topic 1)
All standard backend types support state storage, locking, and remote operations like plan. apply and destroy.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/language/settings/backends/configuration
"Some of these backends act like plain remote disks for state files, while others support locking the state while operations are being performed. This helps prevent conflicts and inconsistencies. The built-in backends listed are the only backends. You cannot load additional backends as plugins."

NEW QUESTION 114
- (Exam Topic 1)
Where in your Terraform configuration do you specify a state backend?

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

Answer: A

Explanation:
Backends are configured with a nested backend block within the top-level terraform block. Reference:
https://www.terraform.io/docs/language/settings/backends/configuration.html https://www.terraform.io/language/settings/backends/configuration#using-a-backend-block

NEW QUESTION 115
- (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 116
- (Exam Topic 1)
How is the Terraform remote backend different than other state backends such as S3, Consul, etc.?

A. It can execute Terraform runs on dedicated infrastructure on premises or in Terraform Cloud
B. It doesn't show the output of a terraform apply locally
C. It is only available to paying customers
D. All of the above

Answer: A

Explanation:
Backends define where Terraform's state snapshots are stored. A given Terraform configuration can either specify a backend, integrate with Terraform Cloud, or do neither and default to storing state locally.
If you and your team are using Terraform to manage meaningful infrastructure, we recommend using the remote backend with Terraform Cloud or Terraform Enterprise.
Reference: https://www.terraform.io/docs/language/settings/backends/index.html

NEW QUESTION 119
- (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 Terrra Terraform language style conventions, along with other minor adjustments for readability.

NEW QUESTION 120
- (Exam Topic 2)
Environment variables can be used to set variables. The environment variables must be in the format " "_<variablename>. Select the correct prefix string from the following list.

A. TF_CLI_ARGS
B. TF_VAR
C. TF_VAR_
D. TF_VAR_ENV

Answer: C

Explanation:
Environment variables can be used to set variables. The environment variables must be in the format TF_VAR_name and this will be checked last for a value. For example:
export TF_VAR_region=us-west-1
export TF_VAR_ami=ami-049d8641 export TF_VAR_alist='[1,2,3]'
export TF_VAR_amap='{ foo = "bar", baz = "qux" }' https://www.terraform.io/docs/commands/environment-variables.html

NEW QUESTION 125
- (Exam Topic 2)
Which of the following represents a feature of Terraform Cloud that is NOT free to customers?

A. Roles and Team Management
B. WorkSpace Management
C. Private Module Registry
D. VCS Integration

Answer: A

Explanation:
Role Based Access Controls (RBAC) for controlling permissions for who has access to what configurations within an organization and it is not free to customers.
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 126
- (Exam Topic 2)
In regards to deploying resources in multi-cloud environments, what are some of the benefits of using Terraform rather than a provider's native tooling? (select

three)

A. Terraform can help businesses deploy applications on multiple clouds and on-premises infrastructure.
B. Terraform is not cloud-agnostic and can be used to deploy resources across a single public cloud.
C. Terraform simplifies management and orchestration, helping operators build large-scale, multi-cloud infrastructure.
D. Terraform can manage cross-cloud dependencies.

Answer: ACD

Explanation:
Terraform is cloud-agnostic and allows a single configuration to be used to manage multiple providers, and to even handle cross-cloud dependencies. This simplifies management and orchestration, helping operators build large-scale multi-cloud infrastructures.
https://www.terraform.io/intro/use-cases.html

NEW QUESTION 131
- (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 135
- (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 136
- (Exam Topic 2)
If you enable TF_LOG = DEBUG, the log will be stored in syslog.log file in the currect directory.

A. False
B. True

Answer: A

Explanation:
https://www.terraform.io/docs/internals/debugging.html

NEW QUESTION 140
- (Exam Topic 2)
How can you ensure that the engineering team who has access to git repo will not create any non-compliant resources that might lead to a security audit failure in future. your team is using Hashicorp Terraform Enterprise Edition.

A. Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using Policy-As-Code.
B. Implement a review process where every code will be reviewed before merging to the master branch.
C. Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and writePolicy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.
D. Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.

Answer: C

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

NEW QUESTION 144
- (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 146
- (Exam Topic 2)
Which of the following clouds does not have a provider maintained HashiCorp?

A. IBM Cloud
B. DigitalOcean
C. OpenStack
D. AWS

Answer: A

Explanation:
IBM Cloud does not have a provider maintained by HashiCorp, although IBM Cloud does maintain their own Terraform provider.
https://www.terraform.io/docs/providers/index.html

NEW QUESTION 149
- (Exam Topic 2)
terraform state subcommands such as list are read-only commands, do read-only commands create state backup files?

A. Yes
B. No

Answer: B

Explanation:
Subcommands that are read-only (such as list) do not write any backup files since they aren't modifying the state.
All terraform state subcommands that modify the state write backup files. The path of these backup file can be controlled with -backup.
https://www.terraform.io/docs/commands/state/index.html#backups

NEW QUESTION 154
- (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 159
- (Exam Topic 2)
lookup retrieves the value of a single element from which of the below data type?

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

Answer: A

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

NEW QUESTION 160
- (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 165
- (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 167
- (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 171
- (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 172
- (Exam Topic 2)
You have declared a variable name my_var in terraform configuration without a value associated with it. variable my_var {}
After running terraform plan it will show an error as variable is not defined.

A. True
B. False

Answer: B

Explanation:
Input variables are usually defined by stating a name, type and a default value. However, the type and default values are not strictly necessary. Terraform can deduct the type of the variable from the default or input value.
Variables can be predetermined in a file or included in the command-line options. As such, the simplest variable is just a name while the type and value are selected based on the input.
variable "variable_name" {}
terraform apply -var variable_name="value"
The input variables, like the one above, use a couple of different types: strings, lists, maps, and boolean. Here are some examples of how each type are defined and used.
String
Strings mark a single value per structure and are commonly used to simplify and make complicated values more user-friendly. Below is an example of a string variable definition.
variable "template" { type = string

default = "01000000-0000-4000-8000-000030080200"
}
A string variable can then be used in resource plans. Surrounded by double quotes, string variables are a simple substitution such as the example underneath.
storage = var.template List
Another type of Terraform variables lists. They work much like a numbered catalogue of values. Each value can be called by their corresponding index in the list. Here is an example of a list variable definition.
variable "users" { type = list
default = ["root", "user1", "user2"]
}
Lists can be used in the resource plans similarly to strings, but you'll also need to denote the index of the value you are looking for.
username = var.users[0] Map
Maps are a collection of string keys and string values. These can be useful for selecting values based on predefined parameters such as the server configuration by the monthly price.
variable "plans" { type = map default = {
"5USD" = "1xCPU-1GB" "10USD" = "1xCPU-2GB" "20USD" = "2xCPU-4GB"
}
}
You can access the right value by using the matching key. For example, the variable below would set the plan to "1xCPU-1GB".
plan = var.plans["5USD"]
The values matching to their keys can also be used to look up information in other maps. For example, underneath is a shortlist of plans and their corresponding storage sizes.
variable "storage_sizes" { type = map
default = {
"1xCPU-1GB" = "25"
"1xCPU-2GB" = "50"
"2xCPU-4GB" = "80"
}
}
These can then be used to find the right storage size based on the monthly price as defined in the previous example.
size = lookup(var.storage_sizes, var.plans["5USD"])
Boolean
The last of the available variable type is boolean. They give the option to employ simple true or false values. For example, you might wish to have a variable that decides when to generate the root user password on a new deployment.
variable "set_password" { default = false
}
The above example boolean can be used similarly to a string variable by simply marking down the correct variable.
create_password = var.set_password
By default, the value is set to false in this example. However, you can overwrite the variable at deployment by assigning a different value in a command-line variable.
terraform apply -var set_password="true"

NEW QUESTION 175
- (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 178
- (Exam Topic 2)
Please identify the offerings which are unique to Terraform Enterprise, and not available in either Terraform OSS, or Terraform Cloud. Select four.

A. Audit Logs
B. Private Network Connectivity
C. VCS Integration
D. Sentinel
E. Clustering

Answer: ABE

Explanation:
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 179
- (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 184
- (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 185
- (Exam Topic 2)
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a _____.

A. First Time Configuration
B. Default Configuration
C. Changing Configuration
D. Partial Configuration
E. Incomplete Configuration

Answer: D

Explanation:
You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.
With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments:
* Interactively: Terraform will interactively ask you for the required values, unless interactive input is disabled. Terraform will not prompt for optional values.
* File: A configuration file may be specified via the init command line. To specify a file, use the
-backend-config=PATH option when running terraform init. If the file contains secrets it may be kept in a secure data store, such as Vault, in which case it must be downloaded to the local disk before running Terraform.
* Command-line key/value pairs: Key/value pairs can be specified via the init command line. Note that many shells retain command-line flags in a history file, so this isn't recommended for secrets. To specify a single key/value pair, use the -backend-config="KEY=VALUE" option when running terraform init.
https://www.terraform.io/docs/backends/config.html#partial-configuration

NEW QUESTION 186
- (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 187
- (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 191
- (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 195
- (Exam Topic 3)
In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

A. 5
B. 2
C. 3
D. 1

Answer: D

Explanation:
A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.
https://www.terraform.io/docs/cloud/workspaces/vcs.html

NEW QUESTION 198
- (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 199
- (Exam Topic 3)
You are reviewing Terraform configurations for a big project in your company. You noticed that there are several identical sets of resources that appear in multiple configurations. What feature of Terraform would you recommend to use to reduce the amount of cloned configuration between the different configurations?

A. Packages
B. Backends
C. Provisioners
D. Modules

Answer: D

Explanation:
Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.
A module is a container for multiple resources that are used together. Modules can be used to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.
Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.
https://www.terraform.io/docs/modules/index.html

NEW QUESTION 202
- (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 205
- (Exam Topic 3)
You have created two workspaces PROD and DEV. You have switched to DEV and provisioned DEV infrastructure from this workspace. Where is your state file stored?

A. terraform.d
B. terraform.tfstate
C. terraform.tfstate.DEV
D. terraform.tfstate.d

Answer: D

Explanation:
Terraform stores the workspace states in a directory called terraform.tfstate.d. This directory should be treated similarly to default workspace state file terraform.tfstate main.tf
provider.tf terraform.tfstate.d DEV
terraform.tfstate # DEV workspace state file PROD
terraform.tfstate # PROD workspace state file terraform.tfvars # Default workspace state file variables.tf

NEW QUESTION 206
- (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 211
- (Exam Topic 3)
Your manager has instructed you to start using terraform for your day-to-day operations, but your security team is concerned about the terraform state files. They have heard it contains confidential information, and are worried that it will not be securely protected. What should be your response to the security team in this regard?

A. Inform the security team that using terraform state is optional . There are ways to avoid it , and you will do the same.
B. Ensure that the state is managed in a remote backend , preferably an enterprise grade state management system like Terraform Cloud.
C. Mask the confidential entries in the terraform state file , using Vault Enterprise, another Hashicorp product , while keeping it locally.
D. Keep the state file locally on each developer machine , and ensure that there is a local protection software like KeyPass protecting it.

Answer: B

Explanation:
https://www.terraform.io/docs/state/index.html
State is very important topic for exam. Please read all of the below subtopics Purpose
Import Existing Resources Locking
Workspaces Remote State Sensitive Data

NEW QUESTION 213
- (Exam Topic 3)
Which of the following value will be accepted for var1? variable "var1" {
type = string
}

A. None of the above
B. Both A and B
C. "5"
D. 5

Answer: B

Explanation:
Terraform automatically converts number and bool values to strings when needed.

NEW QUESTION 215

- (Exam Topic 3)
You have multiple developers working on a terraform project (using terraform OSS), and have saved the terraform state in a remote S3 bucket . However ,team is intermittently experiencing inconsistencies in the provisioned infrastructure / failure in the code . You have traced this problem to simultaneous/concurrent runs of terraform apply command for 2/more developers . What can you do to fix this problem?

A. Use terraform workspaces feature, this will fix this problem by default , as every developer will have their own state file , and terraform will merge them on server side on its own.
B. Structure your team in such a way that only one individual will run terraform apply , everyone will just make changes and share with hi
C. Then there will be no chance of any inconsistencies.
D. Stop using remote state , and store the developer tfstate in their own machine . Once a day , all developers should sit together and merge the state files manually , to avoid any inconsistencies.
E. Enable terraform state locking for the S3 backend using DynamoDB tabl
F. This prevents others from acquiring the lock and potentially corrupting your state.

Answer: D

Explanation:
S3 backend support state locking using DynamoDB. https://www.terraform.io/docs/state/locking.html

NEW QUESTION 218
- (Exam Topic 3)
If you delete a remote backend from the configuration, will you need to rebuild your state files locally?

A. False
B. True

Answer: A

Explanation:
You can change your backend configuration at any time. You can change both the configuration itself as well as the type of backend (for example from "consul" to "s3").
Terraform will automatically detect any changes in your configuration and request a reinitialization. As part of the reinitialization process, Terraform will ask if you'd like to migrate your existing state to the new configuration. This allows you to easily switch from one backend to another.
https://www.terraform.io/docs/backends/config.html#changing-configuration

NEW QUESTION 220
- (Exam Topic 3)
The Security Operations team of ABC Enterprise wants to mandate that all the Terraform configuration that creates an S3 bucket must have encryption feature enabled. What is the best way to achieve it?

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

Answer: A

Explanation:
Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.
Using Sentinel with Terraform Cloud involves:
* Defining the policies - Policies are defined using the policy language with imports for parsing the Terraform plan, state and configuration.
* Managing policies for organizations - Users with permission to manage policies can add policies to their organization by configuring VCS integration or uploading policy sets through the API. They also define which workspaces the policy sets are checked against during runs. (More about permissions.)
* Enforcing policy checks on runs - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is executed.
* Mocking Sentinel Terraform data - Terraform Cloud provides the ability to generate mock data for any run within a workspace. This data can be used with the Sentinel CLI to test policies before deployment.
https://www.terraform.io/docs/cloud/sentinel/index.html

NEW QUESTION 222
- (Exam Topic 3)
Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.

A. False
B. True

Answer: B

Explanation:
Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.
https://www.terraform.io/docs/state/sensitive-data.html#recommendations

NEW QUESTION 227
- (Exam Topic 3)
You cannot publish your own modules on the Terraform Registry.

A. False
B. True

Answer: A

Explanation:
Anyone can publish and share modules on the Terraform Registry. https://www.terraform.io/docs/registry/modules/publish.html

NEW QUESTION 231
- (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 236
- (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 239
- (Exam Topic 3)
When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?

A. Object Storage
B. State Locking
C. WorkSpaces
D. Encryption

Answer: B

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.
State locking happens automatically on all operations that could write state. You won't see any message that it is happening. If state locking fails, Terraform will not continue. You can disable state locking for most commands with the -lock flag but it is not recommended.
If acquiring the lock is taking longer than expected, Terraform will output a status message. If Terraform doesn't output a message, state locking is still occurring if your backend supports it.
Not all backends support locking. Please view the list of backend types for details on whether a backend supports locking or not.
https://www.terraform.io/docs/state/locking.html

NEW QUESTION 244
- (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 245
- (Exam Topic 3)
Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance. Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

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

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.
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.
This example will taint a single resource:
$ terraform taint aws_security_group.allow_all
The resource aws_security_group.allow_all in the module root has been marked as tainted. https://www.terraform.io/docs/commands/taint.html

NEW QUESTION 248
- (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 252
- (Exam Topic 3)
Taint the resource "aws_instance" "baz" resource that lives in module bar which lives in module foo.

A. terraform taint module.foo.module.bar.baz
B. terraform taint module.foo.bar.aws_instance.baz
C. terraform taint module.foo.module.bar.aws_instance.baz
D. terraform taint foo.bar.aws_instance.baz

Answer: C

Explanation:
Check resource addressing https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION 253
- (Exam Topic 3)
You have created an AWS EC2 instance of type t2.micro through your terraform configuration file ec2.tf . Now you want to change the instance type from t2.micro to t2.medium. Accordingly you have changed your configuration file and and ran terraform plan. After running terraform plan you check the output and saw one instance will be updated from t2.micro --> t2.medium. After this you went to grab a coffee without running terraform apply and meanwhile a member of your team changed the instance type of that EC2 instance to t2.medium from aws console. After coming to your desk you run terraform apply. What will happen?

A. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
B. The instance type will be changed to t2.micro and again will be changed to t2.medium
C. terraform apply will through an error.
D. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.

Answer: A

NEW QUESTION 256
- (Exam Topic 3)
Terraform-specific settings and behaviors are declared in which configuration block type?

A. provider

B. terraform
C. resource
D. data

Answer: B

Explanation:
The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.
Example terraform {
required_version = "> 0.12.0"
}
https://www.terraform.io/docs/configuration/terraform.html

NEW QUESTION 260
- (Exam Topic 3)
Multiple providers can be declared within a single Terraform configuration file.

A. True
B. False

Answer: A

Explanation:
You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.
To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration.
For Example
# The default provider configuration provider "aws" {
region = "us-east-1"
}
# Additional provider configuration for west coast region provider "aws" {
alias = "west" region = "us-west-2"
}
The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION 265
- (Exam Topic 3)
Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?

A. Use data source aws_ami.
B. Create a map of region to ami id.
C. Create different configuration file for different region.
D. None of the above

Answer: A

Explanation:
https://www.terraform.io/docs/configuration/data-sources.html

NEW QUESTION 267
- (Exam Topic 3)
A single terraform resource file that defines an aws_instance resource can simply be renamed to vsphere_virtual_machine in order to switch cloud providers.

A. True
B. False

Answer: B

Explanation:
Every provider has its own required and allowed declarations none of which match between cloud providers.

NEW QUESTION 272
- (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 275
- (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 277
- (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 280
- (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 282
- (Exam Topic 3)
During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

A. The resource will be planned for destruction and recreation upon the next terraform apply
B. Terraform will retry to provision again.
C. The failure of provisioner will be ignored and it will not cause a failure to terraform apply
D. The resource will be automatically destroyed.

Answer: A

Explanation:
If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply. Terraform does this because a failed provisioner can leave a resource in a semi-configured state. Because Terraform cannot reason about what the provisioner does, the only way to ensure proper creation of a resource is to recreate it. This is tainting.
You can change this behavior by setting the on_failure attribute, which is covered in detail below. https://www.terraform.io/docs/provisioners/index.html#creation-time-provisioners https://www.terraform.io/docs/provisioners/index.html#destroy-time-provisioners https://www.terraform.io/docs/provisioners/index.html#failure-behavior

NEW QUESTION 285
- (Exam Topic 3)
A user has created three workspaces using the command line - prod, dev, and test. The user wants to create a fourth workspace named stage. Which command will the user execute to accomplish this?

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

Answer: A

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

NEW QUESTION 290
- (Exam Topic 3)
You have provisioned some aws resources in your test environment through Terraform for a POC work. After the POC, now you want to destroy the resources but before destroying them you want to check what resources will be getting destroyed through terraform. what are the options of doing that? (Select TWO)

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

Answer: AD

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

NEW QUESTION 293
- (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 295
- (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 297
- (Exam Topic 4)
You have decided to create a new Terraform workspace to deploy a development environment. What is different about this workspace?

A. It uses a different branch of code It uses a different backend
B. It has its own state file
C. It pulls in a different terraform.tvars file

Answer: C

NEW QUESTION 298
- (Exam Topic 4)
HashiCorp offers multiple versions of Terraform, including Terraform open-source, Terraform Cloud, and Terraform Enterprise. Which of the following Terraform features are only available in the Enterprise edition? (select four)

A. SAML/SSO
B. Sentinel
C. Audit Logs
D. Clustering
E. Private Module Registry
F. Private Network Connectivity

Answer: ACF

Explanation:

While there are a ton of features that are available to open source users, many features that are part of the Enterprise offering are geared towards larger teams and enterprise functionality. To see what specific features are part of Terraform Cloud and Terraform Enterprise, check out this link.
https://www.hashicorp.com/products/terraform/pricing/

NEW QUESTION 302

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

A. True
B. False

Answer: B

Explanation:

https://www.terraform.io/docs/configuration/functions/index.html

NEW QUESTION 304
- (Exam Topic 4)
Select the most accurate statement to describe the Terraform language from the following list.

A. Terraform is an immutable, declarative, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally JSON.
B. Terraform is a mutable, declarative, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
C. Terraform is an immutable, procedural, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.
D. Terraform is a mutable, procedural, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally YAML.

Answer: A

Explanation:
Terraform is not a configuration management tool - https://www.terraform.io/intro/vs/chefpuppet.html Terraform is a declarative language - https://www.terraform.io/docs/configuration/index.html Terraform supports a syntax that is JSON compatible https://www.terraform.io/docs/configuration/syntax-json.html
Terraform is primarily designed on immutable infrastructure principles - https://www.hashicorp.com/resources/what-is-mutable-vs-immutable-infrastructure

NEW QUESTION 308
- (Exam Topic 4)
What does the command terraform fmt do?

A. Rewrite Terraform configuration files to a canonical format and style.
B. Deletes the existing configuration file.
C. Updates the font of the configuration file to the official font supported by HashiCorp.
D. Formats the state file in order to ensure the latest state of resources can be obtained.

Answer: A

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.
Other Terraform commands that generate Terraform configuration will produce configuration files that conform to the style imposed by terraform fmt, so using this style in your own files will ensure consistency.
https://www.terraform.io/docs/commands/fmt.html

NEW QUESTION 312
- (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 314
- (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 317
- (Exam Topic 4)
Which one is the right way to import a local module names consul?

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

Answer: BC

Explanation:
A local path must begin with either ./ or ../ to indicate that a local path is intended, to distinguish from a module registry address.
module "consul" {
source = "./consul"
}

NEW QUESTION 320
- (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 325
- (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 330
- (Exam Topic 4)
Any user can publish modules to the public Terraform Module Registry.

A. True
B. False

Answer: B

NEW QUESTION 335
- (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 340
- (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 345
- (Exam Topic 4)
When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)
* 1. terraform
* 2. {
* 3. required_providers
* 4. {
* 5. aws = "~> 1.2.0"
* 6. }
* 7. }

A. 1.3.1
B. 1.2.3
C. 1.2.9
D. 1.3.0

Answer: BC

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

NEW QUESTION 347
- (Exam Topic 4)
Which of the following arguments are required when declaring a Terraform output?

A. sensitive
B. description
C. default
D. value

Answer: D

NEW QUESTION 349
- (Exam Topic 4)
Which of the following is not a benefit of adopting infrastructure as code?

A. Automation
B. Versioning
C. Reusability of code
D. Interpolation

Answer: D

NEW QUESTION 353
- (Exam Topic 4)
Running terraform fmt without any flags in a directory with Terraform configuration files will check the formatting of those files without changing their contents.

A. True
B. False

Answer:

B

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

NEW QUESTION 356
- (Exam Topic 4)
Given the below resource configuration - resource "aws_instance" "web" { # ... count = 4 }
What does the terraform resource address aws_instance.web refer to?

A. It refers to all 4 web instances , together , for further individual segregation , indexing is required , with a 0 based index.
B. It refers to the last web EC2 instance , as by default , if no index is provided , the last / N-1 index is used.
C. It refers to the first web EC2 instance out of the 4 ,as by default , if no index is provided , the first / 0th index is used.
D. The above will result in a syntax error , as it is not syntactically correct . Resources defined using count , can only be referenced using indexes.

Answer: A

Explanation:
A Resource Address is a string that references a specific resource in a larger infrastructure. An address is made up of two parts:
[module path][resource spec] Module path:
A module path addresses a module within the tree of modules. It takes the form: module.A.module.B.module.C...
Multiple modules in a path indicate nesting. If a module path is specified without a resource spec, the address applies to every resource within the module. If the module path is omitted, this addresses the root module.
Given a Terraform config that includes: resource "aws_instance" "web" {
# ...
count = 4
}
An address like this: aws_instance.web[3]
Refers to only the last instance in the config, and an address like this: aws_instance.web
Refers to all four "web" instances. https://www.terraform.io/docs/internals/resource-addressing.html

NEW QUESTION 361
- (Exam Topic 4)
True or False? terraform init cannot automatically download Community providers.

A. False
B. True

Answer: B

NEW QUESTION 366
- (Exam Topic 4)
When using providers that require the retrieval of data, such as the HashiCorp Vault provider, in what phase does Terraform actually retrieve the data required?

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

Answer: C

NEW QUESTION 369
- (Exam Topic 4)
Which parameters does terraform import require? Choose two correct answers.

A. Provider
B. Path
C. Resource address
D. Resource ID

Answer: CD

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

NEW QUESTION 370
- (Exam Topic 4)
Which task does teraform init not perform?

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

Answer: B

NEW QUESTION 374

- (Exam Topic 4)
How would you be able to reference an attribute from the vsphere_datacenter data source for use with the argument within the vsprere_folder resource in the following configuration?

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

resource "vsphere_folder" "parent" {
          path = "Production"
          type = "vm"
 datacenter id = _________________
}
```

A. vsphere_datacenter.dc.id
B. data.vsphere_datacenter.dc
C. data.dc,id
D. data.vsphere_datacenter.dc.id

Answer: D

NEW QUESTION 376
- (Exam Topic 4)
Terraform variable names are saved in the state file.

A. True
B. False

Answer: B

Explanation:
Terraform stores information about your infrastructure in a state file. This state file keeps track of resources created by your configuration and maps them to real-world resources. https://learn.hashicorp.com/tutorials/terraform/state-cli

NEW QUESTION 381
- (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 385
- (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 388
- (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 390
- (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 394
- (Exam Topic 4)
When should Terraform configuration files be written when running terraform import on existing infrastructure?

A. Infrastructure can be imported without corresponding Terraform code
B. Terraform will generate the corresponding configuration files for you
C. You should write Terraform configuration files after the next terraform import is executed
D. Terraform configuration should be written before terraform import is executed

Answer: D

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

NEW QUESTION 398
- (Exam Topic 4)
A Terraform backend determines how Terraform loads state and stores updates when you execute _____.

A. apply
B. taint
C. destroy
D. All of the above
E. None of the above

Answer: D

NEW QUESTION 400
- (Exam Topic 4)
State is a requirement for Terraform to function

A. True
B. False

Answer: A

Explanation:
State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.
As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).
* 1. Mapping to the Real World
Terraform requires some sort of database to map Terraform config to the real world. When you have a resource resource "aws_instance" "foo" in your configuration, Terraform uses this map to know that instance i- abcd1234 is represented by that resource.
For some providers like AWS, Terraform could theoretically use something like AWS tags. Early prototypes of Terraform actually had no state files and used this method. However, we quickly ran into problems. The first major issue was a simple one: not all resources support tags, and not all cloud providers support tags.
Therefore, for mapping configuration to resources in the real world, Terraform uses its own state structure.
* 2. Metadata
Alongside the mappings between resources and remote objects, Terraform must also track metadata such as resource dependencies.
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.
One way to avoid this would be for Terraform to know a required ordering between resource types. For example, Terraform could know that servers must be deleted before the subnets they are a part of. The
complexity for this approach quickly explodes, however: in addition to Terraform having to understand the ordering semantics of every resource for every cloud, Terraform must also understand the ordering across providers.
Terraform also stores other metadata for similar reasons, such as a pointer to the provider configuration that was most recently used with the resource in situations where multiple aliased providers are present.
* 3. Performance
In addition to basic mapping, Terraform stores a cache of the attribute values for all resources in the state. This is the most optional feature of Terraform state and is done only as a performance improvement.
When running a terraform plan, Terraform must know the current state of resources in order to effectively determine the changes that it needs to make to reach

your desired configuration.
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.
* 4. Syncing
In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but 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.
Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION 405
- (Exam Topic 4)
Resources in terraform can have same identifiers(Resource type + Block name).

A. True
B. False

Answer: B

NEW QUESTION 409
- (Exam Topic 4)
True or False: Workspaces provide identical 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/workspaces/index.html https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION 414
- (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 418
- (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 423
- (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 425
- (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 429
- (Exam Topic 4)
Module version is required to reference a module on the Terraform Module Registry.

A. True
B. False

Answer: B

NEW QUESTION 433
- (Exam Topic 4)
You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed?

A. Terraform can still determine the correct order for destruction from the state even when you delete one or more items from the configuration.
B. Those would be destroyed in the order in which they were written in the configuration file previously before you have deleted them from configuration file.
C. The resource will be destructed in random order as you have already deleted them from configuration.
D. You can not destroy resources by deleting them from configuration file and running plan and apply.

Answer: A

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.

NEW QUESTION 438
- (Exam Topic 4)
Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

A. Unlock
B. force-unlock
C. Removing the lock on a state file is not possible
D. state-unlock

Answer: B

Explanation:

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

NEW QUESTION 439
- (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 444
- (Exam Topic 4)
Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.
What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

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

Answer: D

Explanation:
Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.
This is a great way to slowly transition infrastructure to Terraform.
The terraform import command is used to import existing infrastructure.
To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.
Example:
resource "aws_instance" "import_example" {
# ...instance configuration...
}
Now terraform import can be run to attach an existing instance to this resource configuration.
$ terraform import aws_instance.import_example i-03efafa258104165f aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
aws_instance.import_example: Import complete!
Imported aws_instance (ID: i-03efafa258104165f) aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.
This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside
Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION 447
- (Exam Topic 4)
All Terraform Cloud tiers support team management and governance.

A. True
B. False

Answer: B

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

NEW QUESTION 448
- (Exam Topic 4)
Which of the following is not supported backend types in Terra form?

A. consul
B. gcs
C. manta
D. bitbucket

Answer: D

NEW QUESTION 449
- (Exam Topic 4)
Terraform configuration (including any module references) can contain only one Terraform provider type.

A. True
B. False

Answer: B

NEW QUESTION 452
- (Exam Topic 4)
In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?

A. Resource dependencies are identified and maintained in a file called resource.dependencie
B. Each terraform provider is required to maintain a list of all resource dependencies for the provider and it's included with the plugin during initialization when terraform init is execute
C. The file is located in the terraform.d folder.
D. The terraform binary contains a built-in reference map of all defined Terraform resource dependencies.Updates to this dependency map are reflected in terraform version
E. To ensure you are working with thelatest resource dependency map you much be running the latest version of Terraform.
F. Resource dependencies are handled automatically by the depends_on meta_argument, which is set to true by default.
G. Terraform analyses any expressions within a resource block to find references to other objects, and treats those references as implicit ordering requirements when creating, updating, or destroying resources.

Answer: D

Explanation:

https://www.terraform.io/docs/configuration/resources.html

NEW QUESTION 453
- (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 457
- (Exam Topic 4)
Choose the answer that correctly completes the sentence: _____ backends support state locking.

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

Answer: D

NEW QUESTION 460
- (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 461
- (Exam Topic 4)
Which feature of Terraform allows multiple state files for a single configuration file depending upon the environment?

A. Terraform Modules
B. Terraform Enterprise
C. Terraform Workspaces
D. Terraform Remote Backends

Answer: C

NEW QUESTION 464
- (Exam Topic 4)
Changing the Terraform backend from the default "local" backend to a different one after doing your first terraform apply is:

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

Answer: B

NEW QUESTION 467
- (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 470
- (Exam Topic 4)
A "backend" in Terraform determines how state is loaded and how an operation such as apply is executed. Which of the following is not a supported backend

type?

A. Terraform enterprise
B. Consul
C. Github
D. S3
E. Artifactory

Answer: C

Explanation:
Github is not a supported backend type. https://www.terraform.io/docs/backends/types/index.html

NEW QUESTION 472
- (Exam Topic 4)
Using multi-cloud and provider-agnostic tools provides which of the following benefits?

A. Operations teams only need to learn and manage a single tool to manage infrastructure, regardless of where the infrastructure is deployed.
B. Increased risk due to all infrastructure relying on a single tool for management.
C. Can be used across major cloud providers and VM hypervisors.
D. Slower provisioning speed allows the operations team to catch mistakes before they are applied.

Answer: AC

Explanation:
Using a tool like Terraform can be advantageous for organizations deploying workloads across multiple public and private cloud environments. Operations teams only need to learn a single tool, single language, and can use the same tooling to enable a DevOps-like experience and workflows.

NEW QUESTION 474
- (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 475
- (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 480
- (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 483
- (Exam Topic 4)
How do you specify a module's version when publishing it to the public Terraform Module Registry?

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

Answer: C

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

NEW QUESTION 484
- (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 489
- (Exam Topic 4)
During a terraform plan, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

A. Terraform attempts to provision the resource up to three times before exiting with an error
B. the terraform plan is rolled back and all provisioned resources are removed
C. it is automatically deleted
D. the resource is marked as tainted

Answer: D

Explanation:
If a resource successfully creates but fails during provisioning, Terraform will error and mark the resource as "tainted". A resource that is tainted has been physically created, but can't be considered safe to use since provisioning failed. Terraform also does not automatically roll back and destroy the resource during the apply when the failure happens, because that would go against the execution plan: the execution plan would've said a resource will be created, but does not say it will ever be deleted.

NEW QUESTION 492
- (Exam Topic 4)
You need to specify a dependency manually. What resource meta-parameter can you use lo make sure Terraform respects thee dependency?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
depends_on

NEW QUESTION 493
- (Exam Topic 4)
When you use a remote backend that needs authentication. HashrCorp recommends that you:

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

Answer: C

Explanation:
We recommend omitting the token from the configuration, and instead using terraform login or manually configuring credentials in the CLI config file. Reference:
https://www.terraform.io/language/settings/backends/remote

NEW QUESTION 496
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

Answer: A

NEW QUESTION 497
- (Exam Topic 4)
What advantage does an operations team that uses infrastructure as code have?

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

Answer: B

NEW QUESTION 501
- (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 503
- (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 508
- (Exam Topic 4)
What is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault?

A. Vault provider
B. API access using the AppRole auth method
C. integration with a tool like Jenkins
D. CLI access from the same machine running Terraform

Answer: A

NEW QUESTION 512
- (Exam Topic 4)
Which of the following is not an advantage of using infrastructure as code operations?

A. Self-service infrastructure deployment
B. Troubleshoot via a Linux diff command
C. Public cloud console configuration workflows
D. Modify a count parameter to scale resources
E. API driven workflows

Answer: B

Explanation:
terraform is used to deploy the infrastructure, not to troubleshoot it

NEW QUESTION 515
- (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 520
- (Exam Topic 4)
You need to migrate a workspace to use a remote backend. After updating your configuration, what command do you run to perform the migration?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Once you have authenticated to Terraform Cloud, you're ready to migrate your local state file to Terraform Cloud. To begin the migration, reinitialize. This causes Terraform to recognize your cloud block configuration.

NEW QUESTION 521
- (Exam Topic 4)
In the below configuration, how would you reference the module output vpc_id?

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

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

A. Mastered
B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 526
- (Exam Topic 4)
When Terraform needs to be installed in a location where it does not have internet access to download the installer and upgrades, the installation is generally known as to be _____ .

A. a private install
B. disconnected
C. air-gapped
D. non-traditional

Answer: D

Explanation:

A Terraform Enterprise install that is provisioned on a network that does not have Internet access is generally known as an air-gapped install. These types of installs require you to pull updates, providers, etc. from external sources vs. being able to download them directly.

NEW QUESTION 530
- (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 535
- (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 539
- (Exam Topic 4)
As a member of an operations team that uses infrastructure as code (IaC) practices, you are tasked with making a change to an infrastructure stack running in a public cloud. Which pattern would follow IaC best practices for making a change?

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

Answer: C

NEW QUESTION 542

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

TA-002-P Practice Exam Features:

* TA-002-P Questions and Answers Updated Frequently

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

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

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