Exam Questions TA-002-P

HashiCorp Certified: Terraform Associate

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

NEW QUESTION 1
- (Exam Topic 1)
What command should you run to display all workspaces for the current configuration?

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

Answer: C

Explanation:

terraform workspace list
The command will list all existing workspaces.
Reference: https://www.terraform.io/docs/cli/commands/workspace/list.html

NEW QUESTION 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)
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 4
- (Exam Topic 1)
Which of the following is not a valid string function in Terraform?

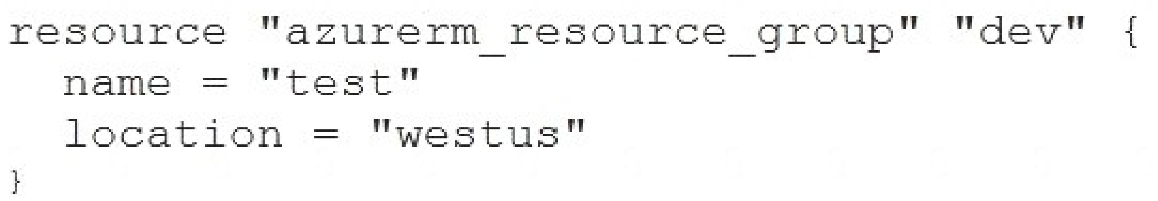
A. split
B. join
C. slice
D. chomp

Answer: C

Explanation:

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

NEW QUESTION 5
- (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 6
- (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 7
- (Exam Topic 1)
Which argument(s) is (are) required when declaring a Terraform variable?

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

Answer: B

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

NEW QUESTION 8
- (Exam Topic 1)
Why would you use the terraform taint command?

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

Answer: B

Explanation:
The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.
Reference: https://www.terraform.io/docs/cli/commands/taint.html

NEW QUESTION 9
- (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 10
- (Exam Topic 1)
You should store secret data in the same version control repository as your Terraform configuration.

A. True
B. False

Answer: B

Explanation:

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

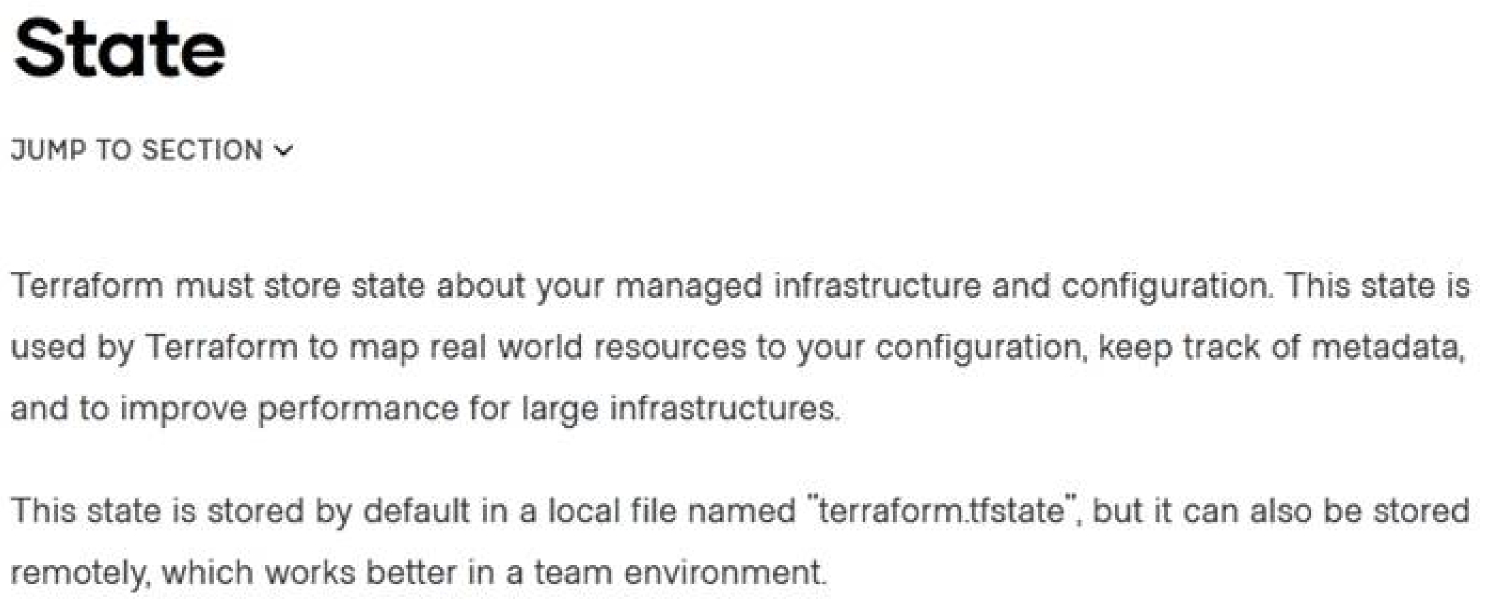
NEW QUESTION 10
- (Exam Topic 1)
FILL BLANK

What is the name of the default file where Terraform stores the state?
Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
"This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment."
https://www.terraform.io/language/state



State

JUMP TO SECTION ⌄

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

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

NEW QUESTION 14
- (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

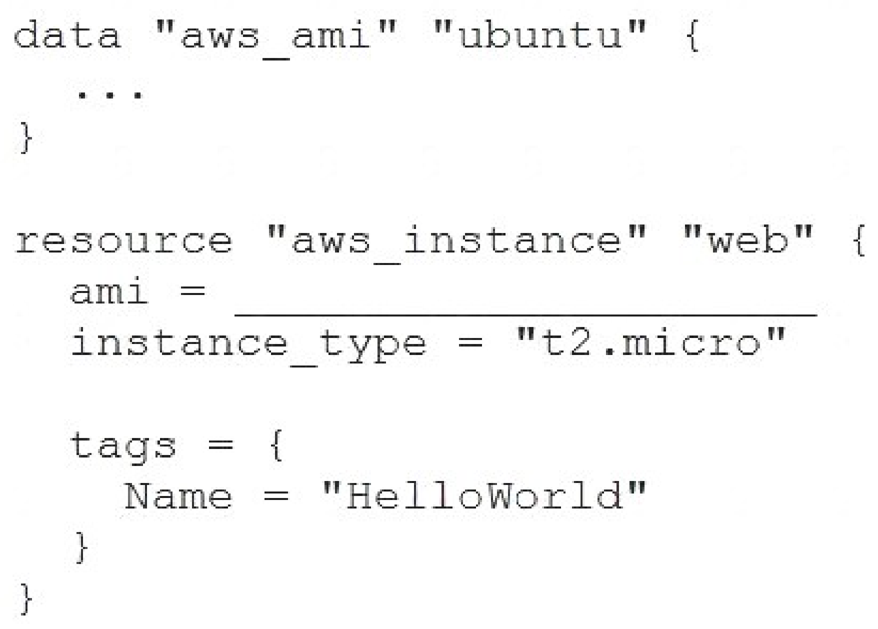
NEW QUESTION 19
- (Exam Topic 1)
Which two steps are required to provision new infrastructure in the Terraform workflow? (Choose two.)

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

Answer: BD

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

NEW QUESTION 20
- (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

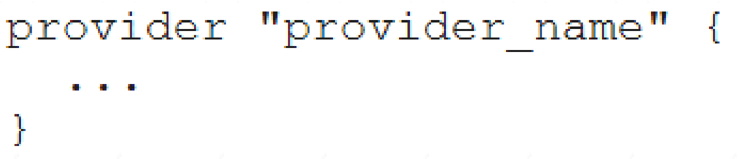
NEW QUESTION 25
- (Exam Topic 1)
What value does the Terraform Cloud/Terraform Enterprise private module registry provide over the public Terraform Module Registry?

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



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

A. True
B. False

Answer: B

Explanation:
Unlike many other objects in the Terraform language, a provider block may be omitted if its contents would otherwise be empty. Terraform assumes an empty default configuration for any provider that is not explicitly configured. https://www.terraform.io/language/providers/configuration

NEW QUESTION 31
- (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 34

- (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 37
- (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 38
- (Exam Topic 1)
You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.
What are the two things you must do to achieve this? (Choose two.)

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

Answer: BC

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

NEW QUESTION 39
- (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 42
- (Exam Topic 1)
Only the user that generated a plan may apply it.

A. True
B. False

Answer: B

NEW QUESTION 46
- (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 49
- (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 53
- (Exam Topic 1)
Terraform can import modules from a number of sources – which of the following is not a valid source?

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

Answer: A

Explanation:
https://www.terraform.io/language/modules/sources

NEW QUESTION 58
- (Exam Topic 1)
Terraform validate reports syntax check errors from which of the following scenarios?

A. Code contains tabs indentation instead of spaces
B. There is missing value for a variable
C. The state files does not match the current infrastructure
D. None of the above

Answer: B

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

NEW QUESTION 62
- (Exam Topic 1)
Which of the following is the correct way to pass the value in the variable num_servers into a module with the input servers?

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

Answer: D

Explanation:
"Within the module that declared a variable, its value can be accessed from within expressions as var.<NAME>, where <NAME> matches the label given in the declaration block:
Note: Input variables are created by a variable block, but you reference them as attributes on an object named var."
https://www.terraform.io/language/values/variables#using-input-variable-values

NEW QUESTION 63
- (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

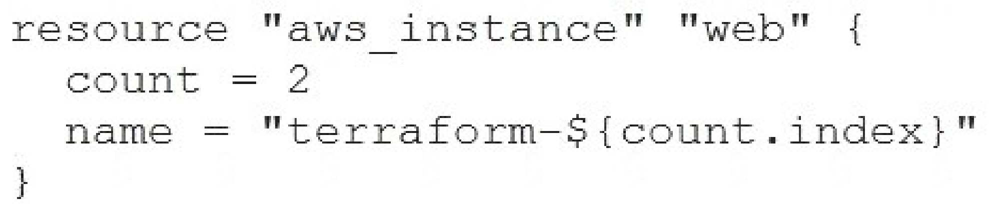
NEW QUESTION 68
- (Exam Topic 1)
If a module uses a local variable, you can expose that value with a terraform output.

A. True
B. False

Answer: A

Explanation:
Output values are like function return values.
Reference: https://www.terraform.io/docs/language/values/locals.html https://www.terraform.io/docs/language/values/outputs.html

NEW QUESTION 72
- (Exam Topic 1)
How would you reference the "name" value of the second instance of this fictitious resource?



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

A. element(aws_instance.web, 2)
B. aws_instance.web[1].name
C. aws_instance.web[1]
D. aws_instance.web[2].name
E. aws_instance.web.*.name

Answer: B

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

NEW QUESTION 75
- (Exam Topic 1)
You need to constrain the GitHub provider to version 2.1 or greater.
Which of the following should you put into the Terraform 0.12 configuration's provider block?

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

Answer: D

Explanation:
version = ">= 1.2.0, < 2.0.0"
A version constraint is a string literal containing one or more conditions, which are separated by commas. Each condition consists of an operator and a version number.
Version numbers should be a series of numbers separated by periods (like 1.2.0), optionally with a suffix to indicate a beta release.
The following operators are valid:
= (or no operator): Allows only one exact version number. Cannot be combined with other conditions.
!=: Excludes an exact version number.
>, >=, <, <=: Comparisons against a specified version, allowing versions for which the comparison is true. "Greater-than" requests newer versions, and "less-than" requests older versions.
~>: Allows only the rightmost version component to increment. For example, to allow new patch releases within a specific minor release, use the full version number: ~> 1.0.4 will allow installation of 1.0.5 and 1.0.10 but not 1.1.0. This is usually called the pessimistic constraint operator.
https://www.terraform.io/language/expressions/version-constraints

NEW QUESTION 80
- (Exam Topic 1)
What is the workflow for deploying new infrastructure with Terraform?

A. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
B. Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to create new infrastructure.
C. terraform plan to import the current infrastructure to the state file, make code changes, and terraform apply to update the infrastructure
D. Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure changes, and terraform apply to create new infrastructure.

Answer: D

Explanation:
Reference: https://www.google.com/search?q=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&oq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new+infrastructure.&aqs=chrome..69i57.556j0j7&sourceid=chrome&ie=UTF-8

NEW QUESTION 82
- (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 85
- (Exam Topic 1)
What is not processed when running a terraform refresh?

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

Answer: B

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

NEW QUESTION 88
- (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 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 92
- (Exam Topic 1)
In Terraform 0.13 and above, outside of the required_providers block, Terraform configurations always refer to providers by their local names.

A. True
B. False

Answer: A

Explanation:
Outside of the required_providers block, Terraform configurations always refer to providers by their local names.
Reference: https://www.terraform.io/docs/language/providers/requirements.html https://www.terraform.io/language/providers/requirements#local-names

NEW QUESTION 93

- (Exam Topic 1)
When using Terraform to deploy resources into Azure, which scenarios are true regarding state files? (Choose two.)

A. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in a new state file
B. When a change is made to the resources via the Azure Cloud Console, Terraform will update the state file to reflect them during the next plan or apply
C. When a change is made to the resources via the Azure Cloud Console, the current state file will not be updated
D. When a change is made to the resources via the Azure Cloud Console, the changes are recorded in the current state file

Answer: BC

NEW QUESTION 98
- (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 102
- (Exam Topic 1)
You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season. Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.
How would you solve this using infrastructure as code?

A. Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource
B. Implement a checklist that engineers can follow when configuring compute instances
C. Replace the compute instance type with a larger version to reduce the number of required deployments
D. Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews

Answer: D

NEW QUESTION 106
- (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 110
- (Exam Topic 1)
Which of the following is not an action performed by terraform init?

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

Answer: A

NEW QUESTION 113
- (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 121
- (Exam Topic 1)
You have multiple team members collaborating on infrastructure as code (IaC) using Terraform, and want to apply formatting standards for readability.
How can you format Terraform HCL (HashiCorp Configuration Language) code according to standard Terraform style convention?

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

Answer: A

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

NEW QUESTION 122
- (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 123
- (Exam Topic 2)
When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

A. False
B. True

Answer: B

Explanation:
TF_LOG_PATH specifies where the log should persist its output to. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
For example, to always write the log to the directory you're currently running terraform from: export TF_LOG_PATH=./terraform.log
export TF_LOG=TRACE

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 127
- (Exam Topic 2)
Provisioners should only be used as a last resort.

A. False
B. True

Answer: B

Explanation:
Provisioners are a Last Resort
Terraform includes the concept of provisioners as a measure of pragmatism, knowing that there will always be certain behaviors that can't be directly represented in Terraform's declarative model.
However, they also add a considerable amount of complexity and uncertainty to Terraform usage. Firstly, Terraform cannot model the actions of provisioners as part of a plan because they can in principle take any action. Secondly, successful use of provisioners requires coordinating many more details than Terraform usage usually requires: direct network access to your servers, issuing Terraform credentials to log in, making sure that all of the necessary external software is installed, etc.
The following sections describe some situations which can be solved with provisioners in principle, but where better solutions are also available. We do not recommend using provisioners for any of the use-cases described in the following sections.
Even if your specific use-case is not described in the following sections, we still recommend attempting to solve it using other techniques first, and use provisioners only if there is no other option.
https://www.terraform.io/docs/provisioners/index.html

NEW QUESTION 132
- (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 133
- (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 134
- (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 135
- (Exam Topic 2)
Terraform init can indeed be run only a few times, because, every time terraform init will initialize the project
, and download all plugins from the internet repository , regardless of whether they were present or not , and this increases the waiting time

A. True
B. False

Answer: B

Explanation:
Re-running init with modules already installed will install the sources for any modules that were added to configuration since the last init, but will not change any already-installed modules. Use -upgrade to override this behavior, updating all modules to the latest available source code.
https://www.terraform.io/docs/commands/init.html

NEW QUESTION 138
- (Exam Topic 2)
The current implementation of Terraform import can only import resources into the state. It does not generate configuration.

A. False
B. True

Answer: B

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

NEW QUESTION 139
- (Exam Topic 2)
Workspaces in Terraform provides similar functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

A. True
B. False

Answer: B

Explanation:
https://www.terraform.io/docs/cloud/migrate/workspaces.html
Workspaces, managed with the terraform workspace command, aren't the same thing as Terraform Cloud's workspaces. Terraform Cloud workspaces act more like completely separate working directories; CLI workspaces are just alternate state files.

NEW QUESTION 141
- (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 146
- (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 151
- (Exam Topic 2)
You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?

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

Answer: C

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

NEW QUESTION 156
- (Exam Topic 2)
By default, a defined provisioner is a creation-time provisioner.

A. True
B. False

Answer: A

Explanation:
https://www.terraform.io/docs/provisioners/index.html

NEW QUESTION 160
- (Exam Topic 2)
You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?

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

Answer: B

Explanation:

Command: force-unlock
Manually unlock the state for the defined configuration.
This will not modify your infrastructure. This command removes the lock on the state for the current configuration. The behavior of this lock is dependent on the backend being used. Local state files cannot be unlocked by another process.
https://www.terraform.io/docs/commands/force-unlock.html https://www.terraform.io/docs/state/locking.html
Terraform has a force-unlock command to manually unlock the state if unlocking failed.
If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed.

NEW QUESTION 165
- (Exam Topic 2)
Which of the following best describes a Terraform provider?

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

Answer: A

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

NEW QUESTION 167
- (Exam Topic 2)
Which of the below configuration file formats are supported by Terraform? (Select TWO)

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

Answer: BE

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

NEW QUESTION 170
- (Exam Topic 2)
Which of the following type of variable allows multiple values of several distinct types to be grouped together as a single value?

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

Answer: BC

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

NEW QUESTION 173
- (Exam Topic 2)
Which of the below terraform commands do not run terraform refresh implicitly before taking actual action of the command?

A. terraform apply
B. terraform destroy
C. terraform init
D. terraform import
E. terraform plan

Answer: CD

Explanation:
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION 178
- (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 182
- (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 185
- (Exam Topic 2)
What does terrafom plan do ?

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

Answer: A

NEW QUESTION 188
- (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 189
- (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 192
- (Exam Topic 3)
You have been given requirements to create a security group for a new application. Since your organization standardizes on Terraform, you want to add this new

security group with the fewest number of lines of code. What feature could you use to iterate over a list of required tcp ports to add to the new security group?

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

Answer: D

Explanation:
A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed value. It iterates over a given complex value and generates a nested block for each element of that complex value.
https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks

NEW QUESTION 196
- (Exam Topic 3)
Your manager has instructed you to start using terraform for the entire infra provisioning of the application stack. There are 4 environments – DEV , QA , UAT , and PROD. The application team has asked for complete segregation between these environments including the backend , state , and also configurations ,since there will be unique resources in different environments . What is the possible way to structure the terraform code to facilitate that.

A. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to a different backend.
B. Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to the same backend.
C. Implement terraform workspaces , and map each environment with one workspace.
D. Enable remote backend storage . Configure 4 different backend storages , one for each environment.

Answer: A

Explanation:
In particular, organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g. staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are not a suitable isolation mechanism for this scenario.
https://www.terraform.io/docs/state/workspaces.html

NEW QUESTION 197
- (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 200
- (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 205
- (Exam Topic 3)
Which of the following is the right substitute for static values that can make Terraform configuration file more dynamic and reusable?

A. Output value
B. Input parameters
C. Functions
D. Modules

Answer: B

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

NEW QUESTION 208
- (Exam Topic 3)
Which of the following state management command allow you to retrieve a list of resources that are part of the state file?

A. terraform state list
B. terraform state view
C. terraform view
D. terraform list

Answer: A

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

NEW QUESTION 213
- (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 217
- (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 222
- (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 227
- (Exam Topic 3)
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. What command will do this?

A. terraform taint
B. terraform apply
C. terraform graph
D. terraform refresh

Answer: A

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 230
- (Exam Topic 3)
Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.

A. Create individual workspaces for each team , and ask them to share modules across workspaces.
B. Implement a Private module registry in Terraform cloud , and ask teams to reference them.
C. Upgrade to Terraform enterprise , since this is not possible in terraform cloud.
D. Upload the modules in the terraform public module registry , and ask teams to reference them

Answer: B

Explanation:
Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster.
By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar.
Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise. Terraform Cloud's private module registry helps you share Terraform modules across your organization.
https://www.terraform.io/docs/cloud/registry/index.html https://www.terraform.io/docs/cloud/registry/publish.html

NEW QUESTION 232
- (Exam Topic 3)
Which of the below features of Terraform can be used for managing small differences between different environments which can act more like completely separate working directories.

A. Repositories
B. Workspaces
C. Environment Variables
D. Backends

Answer: B

Explanation:
workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. They are convenient in a number of situations, but cannot solve all problems.
A common use for multiple workspaces is to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. For example, a developer working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.
Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure. When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that
changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.
https://www.terraform.io/docs/state/workspaces.html https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces

NEW QUESTION 235
- (Exam Topic 3)
Which of the following allows Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code?

A. Sentinel
B. Module registry
C. Functions
D. Workspaces

Answer: A

Explanation:
Sentinel is a language and framework for policy built to be embedded in existing software to enable
fine-grained, logic-based policy decisions. A policy describes under what circumstances certain behaviors are allowed. Sentinel is an enterprise-only feature.
https://www.youtube.com/watch?v=Vy8s7AAvU6g&feature=emb_title

NEW QUESTION 239

- (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 244
- (Exam Topic 3)
State locking does not happen automatically and must be specified at run

A. False
B. True

Answer: A

Explanation:
State locking happens automatically on all operations that could write state. https://www.terraform.io/docs/state/locking.html

NEW QUESTION 245
- (Exam Topic 3)
Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.

A. Set TF_LOG = DEBUG
B. Set VAR_TF = TRACE
C. Set TF_LOG = TRACE
D. Set VAR_TF_LOG = TRACE

Answer: C

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

NEW QUESTION 246
- (Exam Topic 3)
In regards to Terraform state file, select all the statements below which are correct?

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

Answer: ACEF

Explanation:
Terraform state can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.
When using local state, state is stored in plain-text JSON files.
When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.
Storing Terraform state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.
Recommendations
If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.
Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.
For example:
* Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.
* The S3 backend supports encryption at rest when the encrypt option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

NEW QUESTION 248
- (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 250
- (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 253
- (Exam Topic 3)
After creating a new workspace "PROD" you need to run the command terraform select PROD to switch to it.

A. False
B. True

Answer: A

Explanation:
By default, when you create a new workspace you are automatically switched to it
To create a new workspace and switch to it, you can use terraform workspace new <new_workspace_name>; to switch to a existing workspace you can use terraform workspace select <existing_workspace_name>;
Example:
$ terraform workspace new example
Created and switched to workspace "example"!
You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.

NEW QUESTION 254
- (Exam Topic 3)
Which of the below command will upgrade the provider version to the latest acceptable one?

A. terraform plan upgrade
B. terraform provider -upgrade
C. terraform init -upgrade
D. terraform init -update

Answer: C

Explanation:
To upgrade to the latest acceptable version of each provider, run terraform init -upgrade. This command also upgrades to the latest versions of all Terraform modules.
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION 258
- (Exam Topic 3)
Which of the below commands will rename a EC2 instance without destroying and recreating it?

A. terraform state mv
B. terraform mv
C. terraform plan

D. terraform plan mv

Answer: A

NEW QUESTION 263
- (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 268
- (Exam Topic 3)
Which of the below datatype is not supported by Terraform.

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

Answer: A

NEW QUESTION 270
- (Exam Topic 3)
A data block requests that Terraform read from a given data source and export the result under the given local name.

A. False
B. True

Answer: B

NEW QUESTION 271
- (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 272
- (Exam Topic 3)
Terraform Enterprise currently supports running under which the following operating systems?

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

Answer: ABCDEF

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

NEW QUESTION 277
- (Exam Topic 3)
You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.

A. Run the command export TF_LOG_FILE=./terraform.log.
B. Run the command export TF_LOG_PATH=./terraform.log.
C. Run the command export TF_DEBUG_PATH=./terraform.log.
D. No explicit action require
E. Terraform will take care of this as you have enable TF_LOG.

Answer: B

Explanation:
https://www.terraform.io/docs/commands/environment-variables.html

NEW QUESTION 281
- (Exam Topic 3)
Command terraform refresh will update state file?

A. False
B. True

Answer: B

Explanation:
The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. This can be used to detect any drift from the last-known state, and to update the state file.
This does not modify infrastructure, but does modify the state file. If the state is changed, this may cause changes to occur during the next plan or apply.
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION 283
- (Exam Topic 3)
You also have a defined the following environment variables in your shell: TF_itemNum =6, TF_VAR_itemNum =9. You also have a terraform.tfvars file with the following contents
itemNum = 7
When you run the following apply command, what is the value assigned to the itemNum variable? terraform apply -var itemNum =4

A. 10
B. 6
C. 1
D. 4
E. 3

Answer: D

Explanation:
The -var and -var-file methods of assigning variables have the highest precedence. https://www.terraform.io/docs/configuration/variables.html

NEW QUESTION 286
- (Exam Topic 3)
Hanah is writing a terraform configuration with nested modules, there are multiple places where she has to use the same conditional expression but she wants to avoid repeating the same values or expressions multiple times in the configuration,. What is a better approach to dealing with this?

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

Answer: B

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

NEW QUESTION 291
- (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 296

- (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 297
- (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 301
- (Exam Topic 3)
What does terraform refresh command do?

A. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
B. terraform refresh command basically updates the configuration file with the current state of the actual infrastructure
C. terraform refresh is use to change/modify the infrastructure based on the existing state file, at that moment.
D. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
E. terraform refresh syncs the state file with the real world infrastructure.

Answer: E

NEW QUESTION 305
- (Exam Topic 3)
Multiple configurations for the same provider can be used in a single configuration file.

A. False
B. True

Answer: B

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#alias-multiple-provider-instances

NEW QUESTION 309
- (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 310
- (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 312
- (Exam Topic 4)
terraform init retrieves the source code tot all referenced modules

A. True
B. False

Answer: A

Explanation:
Terraform installs providers, initialises source code & modules etc at this stage

NEW QUESTION 313
- (Exam Topic 4)
How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? (Choose two.)

A. A full audit trail of the request and fulfillment process is generated
B. A request must be submitted for infrastructure changes
C. As additional resources are required, more tickets are submitted
D. A catalog of approved resources can be accessed from drop down lists in a request form

Answer: BC

NEW QUESTION 315
- (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 316
- (Exam Topic 4)
A user has created a module called "my_test_module" and committed it to GitHub. Over time, several commits have been made with updates to the module, each tagged in GitHub with an incremental version number. Which of the following lines would be required in a module configuration block in terraform to select tagged version v1.0.4?

A. source = "git::https://example.com/my_test_module.git@tag=v1.0.4"
B. source = "git::https://example.com/my_test_module.git&ref=v1.0.4"
C. source = "git::https://example.com/my_test_module.git#tag=v1.0.4"
D. source = "git::https://example.com/my_test_module.git?ref=v1.0.4"

Answer: D

Explanation:
https://www.terraform.io/docs/modules/sources.html#selecting-a-revision

NEW QUESTION 321
- (Exam Topic 4)
As a developer, you want to ensure your plugins are up to date with the latest versions. Which Terraform command should you use?

A. terreform providers- upgrade

B. terreform apply -upgrade
C. terreform refresh -upgrade
D. terreformn Init -upgrade

Answer: D

NEW QUESTION 325
- (Exam Topic 4)
What does the command terraform fmt do?

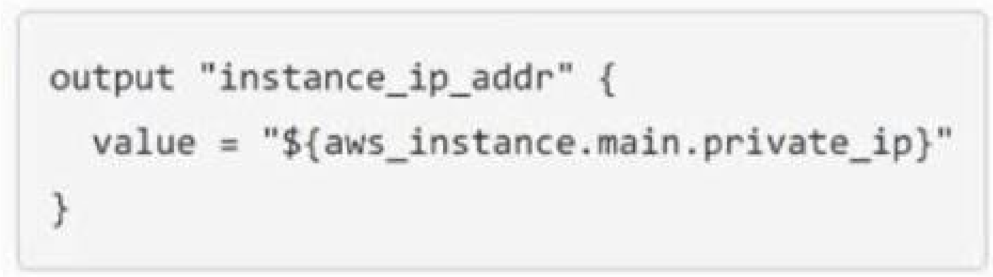
A. Rewrite Terraform configuration files to a canonical format and style.
B. Deletes the existing configuration file.
C. Updates the font of the configuration file to the official font supported by HashiCorp.
D. Formats the state file in order to ensure the latest state of resources can be obtained.

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

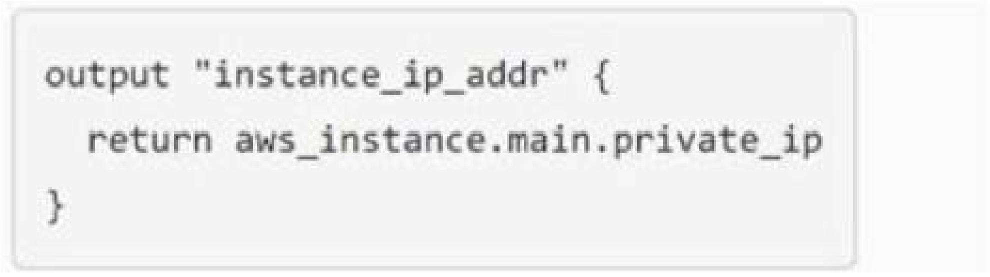
NEW QUESTION 329
- (Exam Topic 4)
You are writing a child Terraform module which provisions an AWS instance. You want to make use of the IP address returned in the root configuration. You name the instance resource "main".
Which of these is the correct way to define the output value using HCL2?

A.



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

B.



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

A. Option A
B. Option B

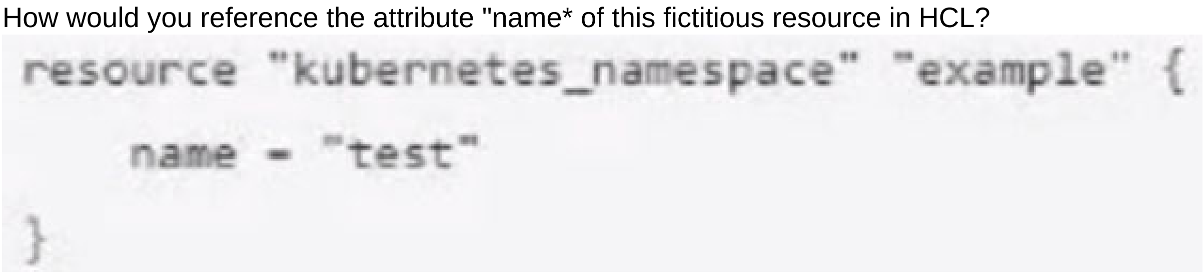
Answer: A

NEW QUESTION 332
- (Exam Topic 4)
In the example below, the depends_on argument creates what type of dependency?

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

Answer: C

NEW QUESTION 334
- (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 339
- (Exam Topic 4)
True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.

A. False
B. True

Answer: A

Explanation:

Currently, Terraform has no mechanism to redact or protect secrets that are returned via data sources, so secrets read via this provider will be persisted into the Terraform state, into any plan files, and in some cases in the console output produced while planning and applying. These artifacts must, therefore, all be protected accordingly.

NEW QUESTION 342
- (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 344
- (Exam Topic 4)
Your risk management organization requires that new AWS S3 buckets must be private and encrypted at rest. How can Terraform Enterprise automatically and proactively enforce this security control?

A. With a Sentinel policy, which runs before every apply
B. By adding variables to each TFE workspace to ensure these settings are always enabled
C. With an S3 module with proper settings for buckets
D. Auditing cloud storage buckets with a vulnerability scanning tool

Answer: A

Explanation:
https://docs.hashicorp.com/sentinel/intro/what
https://medium.com/hashicorp-engineering/enforcing-aws-s3-security-best-practice-using-terraform-sentinel-dd

NEW QUESTION 345
- (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 346
- (Exam Topic 4)

Terra form installs its providers during which phase?

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

Answer: B

Explanation:
Providers are installed in the init phase

NEW QUESTION 347
- (Exam Topic 4)
Any user can publish modules to the public Terraform Module Registry.

A. True
B. False

Answer: B

NEW QUESTION 351
- (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 352
- (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 354
- (Exam Topic 4)
Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)

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

Answer: ABC

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

NEW QUESTION 357
- (Exam Topic 4)
Which of the following statements about Terraform modules is not true?

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

Answer: A

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

NEW QUESTION 361
- (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 363
- (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 368
- (Exam Topic 4)
Which of the following statements best describes the Terraform list(...) type?

A. a collection of values where each is identified by a string label.
B. a sequence of values identified by consecutive whole numbers starting with zero.
C. a collection of unique values that do not have any secondary identifiers or ordering.
D. a collection of named attributes that each have their own type.

Answer: B

Explanation:
A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.
https://www.terraform.io/docs/configuration/types.html#structural-types

NEW QUESTION 373
- (Exam Topic 4)
True or False? terraform init cannot automatically download Community providers.

A. False
B. True

Answer: B

NEW QUESTION 375
- (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 377
- (Exam Topic 4)
You just upgraded the version of a provider in an existing Terraform project. What do you need to do to install the new provider?

A. Run terraform apply -upgrade
B. Run terraform init -upgrade
C. Run terraform refresh
D. Upgrade your version of Terraform

Answer: B

Explanation:
[-upgrade] - Opt to upgrade modules and plugins as part of their respective installation steps. See the sections below for more details. Reference: https://www.terraform.io/cli/commands/init#upgrade

NEW QUESTION 378
- (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)
You're preparing to install Terraform on client workstations and want to see which operating systems are supported. Which of the following operating systems is supported?

A. Windows
B. Amazon Linux
C. FreeBSD
D. Solaris
E. MacOS
F. All of the above

Answer: F

NEW QUESTION 383
- (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 387
- (Exam Topic 4)
You have created a main.tr Terraform configuration consisting of an application server, a database, and a load balancer. You ran terraform apply and all resources were created successfully. Now you realize that you do not actually need the load balancer so you run terraform destroy without any flags What will happen?

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

Answer: B

NEW QUESTION 389
- (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 391
- (Exam Topic 4)
You have to initialize a Terraform backend before it can be configured.

A. True
B. False

Answer: A

Explanation:

Initialization
Whenever a configuration's backend changes, you must run terraform init again to validate and configure the backend before you can perform any plans, applies, or state operations.
When changing backends, Terraform will give you the option to migrate your state to the new backend. This lets you adopt backends without losing any existing state.
To be extra careful, we always recommend manually backing up your state as well. You can do this by simply copying your terraform.tfstate file to another location. The initialization process should create a backup as well, but it never hurts to be safe!
https://www.terraform.io/language/settings/backends/configuration

NEW QUESTION 395
- (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 399
- (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 402
- (Exam Topic 4)
terraform destroy is the only way to remove infrastructure.

A. True
B. False

Answer: B

NEW QUESTION 403
- (Exam Topic 4)
Resources in terraform can have same identifiers(Resource type + Block name).

A. True
B. False

Answer: B

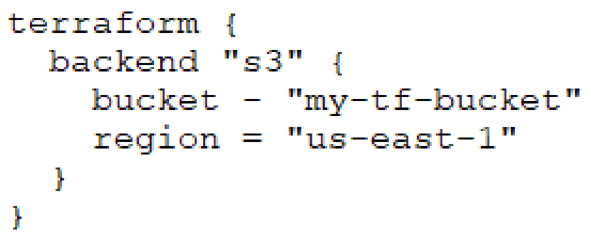
NEW QUESTION 406
- (Exam Topic 4)
What is the result of the following terraform function call?

A. hello
B. what?
C. goodbye

Answer: B

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

NEW QUESTION 409
- (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 412
- (Exam Topic 4)
You have created a custom variable definition file my_vars.tfvars. How will you use it for provisioning infrastructure?

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

Answer: D

Explanation:
To set lots of variables, it is more convenient to specify their values in a variable definitions file (with a filename ending in either .tfvars or .tfvars.json) and then specify that file on the command line with -var-file:
terraform apply -var-file="my_vars.tfvars" https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files

NEW QUESTION 416
- (Exam Topic 4)
colleagues is new toTerraform and wants to add a new workspace named new-hire. What command he should execute from the following?

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

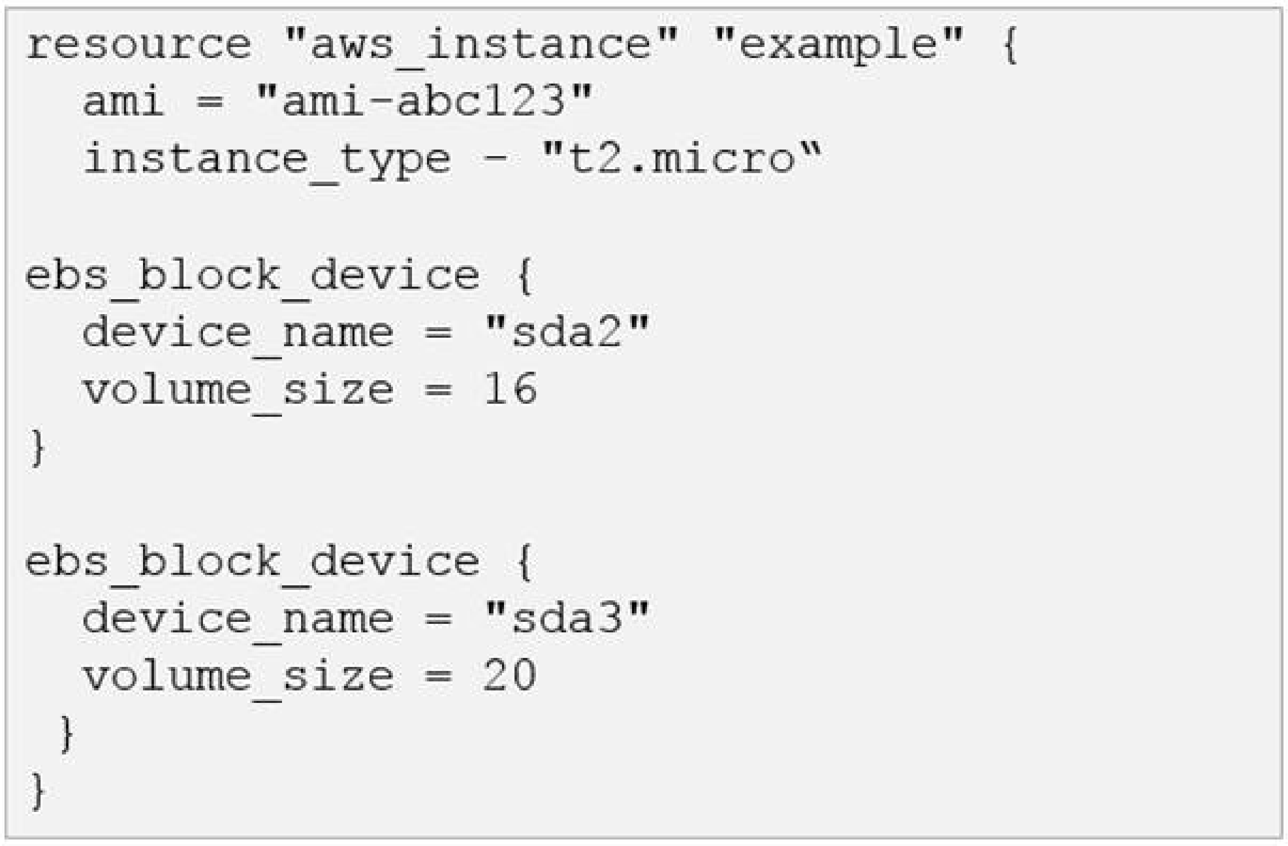
Answer: B

NEW QUESTION 421
- (Exam Topic 4)
Do terraform workspaces help in adding/allowing multiple state files for a single configuration?

A. True
B. False

Answer: A

NEW QUESTION 424
- (Exam Topic 4)
How would you reference the Volume IDs associated with the ebs_block_device blocks in this configuration?



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

ebs_block_device {
  device_name = "sda2"
  volume_size = 16
}

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

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

Answer: A

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

NEW QUESTION 429
- (Exam Topic 4)
What feature of Terraform Cloud and/or Terraform Enterprise can you publish and maintain a set of custom modules which can be used within your organization?

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

Answer: C

NEW QUESTION 434
- (Exam Topic 4)
Your company has a lot of workloads in AWS , and Azure that were respectively created using CloudFormation , and AzureRM Templates. However , now your CIO has decided to use Terraform for all new projects , and has asked you to check how to integrate the existing environment with terraform code. What should be your next plan of action?

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

Answer: B

NEW QUESTION 438
- (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 440
- (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 443
- (Exam Topic 4)
Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?

A. False
B. True

Answer: B

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

NEW QUESTION 445
- (Exam Topic 4)
You have modified your local Terraform configuration and ran terraform plan to review the changes. Simultaneously, your teammate manually modified the infrastructure component you are working on. Since you already ran terraform plan locally, the execution plan for terraform apply will be the same.

A. True
B. False

Answer: B

NEW QUESTION 450
- (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 455
- (Exam Topic 4)
Terraform configuration (including any module references) can contain only one Terraform provider type.

A. True
B. False

Answer: B

NEW QUESTION 458
- (Exam Topic 4)
Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)

A. GitHub

B. CVS Version Control
C. Azure DevOps Server
D. Bitbucket Cloud
E. GitHub Enterprise

Answer: ACDE

Explanation:

Terraform Cloud supports the following VCS providers:
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github.html
- https://www.terraform.io/docs/cloud/vcs/github-enterprise.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-com.html
- https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html
- https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html
- https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers

NEW QUESTION 460
- (Exam Topic 4)
Which of the following is a meta-argument defined in the configuration files of Terraform?

A. tfvar
B. depends_on
C. instance aws
D. varl

Answer: B

NEW QUESTION 463
- (Exam Topic 4)
You're writing a Terraform configuration that needs to read input from a local file called id_rsa.pub. Which built-in Terraform function can you use to import the file's contents as a string?

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

Answer: D

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

NEW QUESTION 468
- (Exam Topic 4)
Which of the following is considered a Terraform plugin?

A. Terraform language
B. Terraform tooling
C. Terraform logic
D. Terraform provider

Answer: D

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.
https://www.terraform.io/docs/plugins/basics.html

NEW QUESTION 473
- (Exam Topic 4)
Select the feature below that best completes the sentence:
The following list represents the different types of ______ available in Terraform.
* 1. max
* 2. min
* 3. join
* 4. replace
* 5. list
* 6. length
* 7. range

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

Answer: D

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

NEW QUESTION 476
- (Exam Topic 4)
If a DevOps team adopts AWS Cloud Formation as their standardized method for provisioning public cloud resources, which of the following scenarios poses a challenge for this team?

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

Answer: B

NEW QUESTION 479
- (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 484
- (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 486
- (Exam Topic 4)
Terraform plan updates your state file.

A. True
B. False

Answer: B

Explanation:
The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure. 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. Source: https://www.terraform.io/cli/commands/plan

NEW QUESTION 491
- (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 494
- (Exam Topic 4)
Consider the following Terraform 0.12 configuration snippet:

```
* 1. variable "vpc_cidrs" {
* 2. type = map
* 3. default = {
* 4. us-east-1 = "10.0.0.0/16"
* 5. us-east-2 = "10.1.0.0/16"
* 6. us-west-1 = "10.2.0.0/16"
* 7. us-west-2 = "10.3.0.0/16"
* 8. }
* 9. }
* 10.
* 11. resource "aws_vpc" "shared" {
* 12. cidr_block = ______
```

* 13. }
How would you define the cidr_block for us-east-1 in the aws_vpc resource using a variable?

A. var.vpc_cidrs.0
B. vpc_cidrs[“us-east-1”]
C. var.vpc_cidrs[“us-east-1”]
D. var.vpc_cidrs[0]

Answer: C

NEW QUESTION 497
- (Exam Topic 4)
You have a simple Terraform configuration containing one virtual machine (VM) in a cloud provider. You run terraform apply and the VM is created successfully. What will happen if you terraform apply again immediately afterwards without changing any Terraform code?

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

Answer: D

NEW QUESTION 501
- (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 506
- (Exam Topic 4)
You want to share Terraform state with your team, store it securely and provide state locking. How would you do this? Choose three correct answers.

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

Answer: ABE

NEW QUESTION 508
- (Exam Topic 4)
When do you need to explicitly execute terraform refresh?

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

Answer: D

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

NEW QUESTION 510
- (Exam Topic 4)
Which of the following is true about terraform apply? (Choose two.)

A. It only operates on infrastructure defined in the current working directory or workspace
B. You must pass the output of a terraform plan command to it
C. Depending on provider specification, Terraform may need to destroy and recreate your infrastructure resources
D. By default, it does not refresh your state file to reflect current infrastructure configuration
E. You cannot target specific resources for the operation

Answer: AC

Explanation:
https://www.terraform.io/cli/run

NEW QUESTION 514
- (Exam Topic 4)
When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)

A. directly querying HashiCorp Vault for the secrets
B. command-line key/value pairs
C. use the -backend-config=PATH to specify a separate config file
D. interactively on the command line

Answer: BCD

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: https://www.terraform.io/docs/backends/init.html#backend-initialization

NEW QUESTION 519
- (Exam Topic 4)
Complete the following sentence:
The terraform state command can be used to ______

A. modify state
B. view state
C. refresh state
D. There is no such command

Answer: A

Explanation:

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

NEW QUESTION 520
- (Exam Topic 4)
Which of the following is the safest way to inject sensitive values into a Terraform Cloud workspace?

A. Write the value to a file and specify the file with the -var-file flag
B. Set a value for the variable in the UI and check the "Sensitive" check box
C. Edit the state file directly just before running terraform apply
D. Set the variable value on the command line with the -var flag

Answer: B

Explanation:
-var and -var-file overwrite workspace-specific and variable set variables that have the same key. From the workspace, variable can be added and checked off as being sensitive. Reference: https://www.terraform.io/cloud-docs/workspaces/variables/managing-variables#loading-variables-from-files
https://www.terraform.io/cloud-docs/workspaces/variables

NEW QUESTION 524
- (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 527
- (Exam Topic 4)
You want to define multiple data disks as nested blocks inside the resource block for a virtual machine. What Terraform feature would help you define the blocks using the values in a variable?

A. Local values
B. Dynamic blocks
C. Count arguments

D. Collection functions

Answer: B

NEW QUESTION 528
- (Exam Topic 4)
terraform validate reports HCL syntax errors.

A. True
B. False

Answer: A

NEW QUESTION 532
- (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 534
- (Exam Topic 4)
What does state locking accomplish?

A. Copies the state file from memory to disk
B. Encrypts any credentials stored within the state file
C. Blocks Terraform commands from modifying the state file
D. Prevents accidental deletion of the state file

Answer: C

Explanation:
If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state. Source: https://www.terraform.io/language/state/locking

NEW QUESTION 535
- (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 536
- (Exam Topic 4)
You have a Terraform configuration that defines a single virtual machine with no references to it. You have run terraform apply to create the resource, and then removed the resource definition from your Terraform configuration file.
What will happen when you run terraform apply in the working directory again?

A. Nothing
B. Terraform will destroy the virtual machine
C. Terraform will error
D. Terraform will remove the virtual machine from the state file, but the resource will still exist

Answer: B

Explanation:
If you remove the resource from your config file and the resource is in your state file, terraform will apply the configuration in the config file - which is to delete the resource

NEW QUESTION 537
- (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 539

......