Which of the following variable definition files will terraform load automatically?

YOUR ANSWER: Any files with names ending in .auto.tfvars.json, terraform.tfvars.json
CORRECT ANSWER: Any files with names ending in .auto.tfvars.json, terraform.tfvar, terraform.tfvars
EXPLANATION: Terraform 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.

What feature of Terraform Cloud 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

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 False
B True

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

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 -create stage
C terraform workspace create stage
D terraform workspace new stage

Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

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

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 refresh
B terraform taint
C terraform apply
D terraform graph

When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)

```
terraform {

required_providers {

aws = “~> 1.2.0 }

}
```

A 1.3.1
B 1.2.9
C 1.2.3
D 1.3.0

Which of the below command will upgrade the provider version to the latest acceptable one ?

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

By default, Terraform destroy will prompt for confirmation before proceeding.

YOUR ANSWER: True
EXPLANATION: Terraform destroy will always prompt for confirmation before executing unless passed the -auto-approve flag. $ terraform destroy Do you really want to destroy all resources? Terraform will destroy all your managed infrastructure, as shown above. There is no undo. Only 'yes' will be accepted to confirm.

Multiple provider instances blocks for AWS can be part of a single configuration file?

A True
B False

Which of the following is valid interpolation syntax for retrieving a data source attribute?

A azurerm_resource_group.test.data
B aws_instance.web.id.data
C data.google_container_cluster.my_cluster.endpoint
D data.google_storage_bucket.backend

In terraform, most resource dependencies are handled automatically. Which of the following statements describes best how terraform resource dependencies are handled?

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

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

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

The terraform import command can import resources directly into Terraform state

A True
B False

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

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

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

What is the result of the following terraform function call?

```
> index(["a", "b", "c"], "b")
```

A True
B 1
C 0
D 2

EXPLANATION: index finds the element index for a given value in a list. index(list, value) The returned index is zero-based. This function produces an error if the given value is not present in the list.

Terraform must track metadata such as resource dependencies. Where is this data stored?

A Workspace
B Metadata store
C Backend
D State file

our team uses terraform OSS . You have created a number of reusable modules for important , independent network components that you want to share with your team to enhance consistency . What is the correct option to do that?

A Terraform modules cannot be shared in OSS version . Each developer needs to maintain their own modules , and leverage them in the main tf file.
B Terraform module sharing is only available in Enterprise version via terraform private module registry , so no way to enable it in OSS version.
C Store your modules in a NAS/ shared file server , and ask your team members to directly reference the code from there . This is the only viable option in terraform OSS ,which is better than individually maintaining module versions for every developer
D Upload your modules with proper versioning in the terraform public module registry . Terraform OSS is directly integrated with the public module registry , and can reference the modules from the code in the main tf file.

You are reviewing Terraform configurations for a big project in your company. You noticed that there are several identical sets of resources that appear in multiple configurations. What feature of Terraform would you recommend to use to reduce the amount of cloned configuration between the different configurations?

A Backends
B Packages
C Provisioners
D Modules

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

During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

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

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

A True
B False

Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements. (select three)

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

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

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 Github
C Artifactory
D S3
E Consul

Terraform has detailed logs which can be enabled by setting the _________ environmental variable.

A TF_TRACE
B TF_INFO
C TF_DEBUG
D TF_LOG

Environment variables can be used to set variables. The environment variables must be in the format "____"_<variablename>. Select the correct prefix string from the following list.

A TF_ENV
B TF_VAR
C TF_ENV_VAR
D TF_VAR_NAME

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

A True
B False

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

A 4
B 0
C 2
D 1

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

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

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 Dynamic block
C Terraform import
D Splat expression

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

A No
B Yes
After creating a new workspace "PROD" you need to run the command terraform select PROD to switch to it.

A True
B False

When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

A False
B True

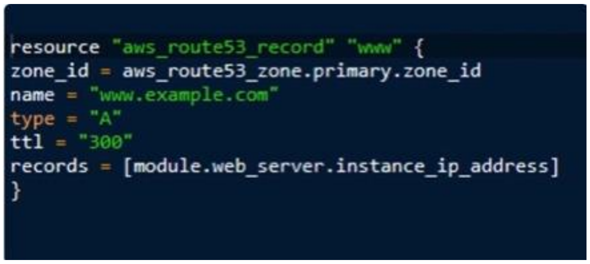
What is the purpose of using the local-exec provisioner? (select two)

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

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

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

In the example below, where is the value of the DNS record's IP address originating from?



```
resource "aws_route53_record" "www" {
zone_id = aws_route53_zone.primary.zone_id
name = "www.example.com"
type = "A"
ttl = "300"
records = [module.web_server.instance_ip_address]
}
```

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

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

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

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

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

A False
B True

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

Question Image
A To ensure that the provider version matches the version of Terraform you are using
B To remove older versions of the provider
C Providers are released on a separate schedule from Terraform itself; therefore a newer version could introduce breaking changes

D To match the version number of your application being deployed via Terraform

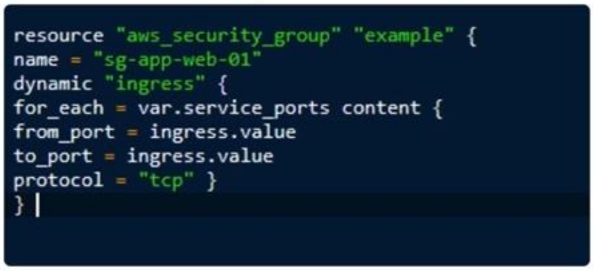
What is the default backend for Terraform?

A Default
B Local
C Consul
D S3

Select two answers to complete the following sentence: Before a new provider can be used, it must be ______ and _______.

A initialized
B approved by HashiCorp
C declared in the configuration
D uploaded to source control

What Terraform feature is shown in the example below?



```
resource "aws_security_group" "example" {
name = "sg-app-web-01"
dynamic "ingress" {
for_each = var.service_ports content {
from_port = ingress.value
to_port = ingress.value
protocol = "tcp" }
} |
```

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

What is the purpose of using the local-exec provisioner?

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

In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to use what?

A output value
B regular expressions
C input parameters
D module

Which of the following best describes the default local backend?

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

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 Workspaces
C Functions
D Module registry

Which of the following represents a feature of Terraform Cloud that is NOT free to customers?

A roles and team management
B private module registry
C workspace management
D VCS integration

What happens when a terraform plan is executed?

A Reconciles the state Terraform knows about with the real-world infrastructure
B Applies the changes required in the target infrastructure in order to reach the desired configuration
C Creates an execution plan and determines what changes are required to achieve the desired state in the configuration files.
D The backend is initialized and the working directory is prepped

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?

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

You have been asked to review Terraform configurations for multiple projects in your company. You noticed that there are several identical sets of resources that appear in multiple projects. What feature of Terraform would you recommend to use to reduce the amount of cloned configuration between the projects?

A Packages
B Backends
C Provisioners
D Modules

Which Terraform command will force a marked resource to be destroyed and recreated on the next apply?

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

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

A Run terraform refresh to ensure that the state file has the latest information for existing resources.
B Update the configuration file to include the new resources
C Shut down or stop using the resources being imported so no changes are inadvertently missed
D Modify the Terraform state file to add the new resources

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

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

What is a downside to using the Vault provider to read secrets from Vault?

A Secrets are persisted to the state file and plans
B Terraform requires a unique auth method to work with Vault
C Terraform and Vault must be running on the same version
D Terraform and Vault must be running on the same physical host

Which of the following connection types are supported by the remote-exec provisioner?

A smb
B rdp
C ssh
D winrm

When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state doesn't become corrupt?

A Encryption
B State locking
C Workspaces
D Object storage

State is a requirement for Terraform to function.

A True
B False

By default, where does Terraform store its state file?

A Remotely using Terraform Cloud
B Current working directory
C Shared directory
D Amazon S3 bucket

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 interactively on the command line
C use the -backend-config=PATH to specify a separate config file
D command-line key/value pairs

From the answers below, select the advantages of using Infrastructure as Code.

A Easily change and update existing infrastructure
B Provide a codified workflow to develop customer-facing applications
C Safely test modifications using a "dry run" before applying any actual changes
D Easily integrate with application workflows (GitLab Actions, Azure DevOps, CI/CD tools)
E Provide reusable modules for easy sharing and collaboration

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

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 Named values
C Functions
D Data sources

Command terraform refresh will update state file?

A True
B False

What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

A Providers
B Remote Backends
C Local Backends
D Workspaces

Complete the following sentence: For local state, the workspaces are stored directly in a...

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

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"

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 plan
B terraform delete
C terraform apply
D terraform init

If you delete a remote backend from the configuration you will need to rebuild your state files locally.

A True
B False

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

A colleague has informed you that a new version of a Terraform module that your team hosts on an Amazon S3 bucket is broken. The Amazon S3 bucket has versioning enabled. Your colleague tells you to make sure you are not using the latest version in your configuration. You have the following configuration block in your code that refers to the below module. Of the available choices, what is the best way to ensure that you are not using the latest version of the module?

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

State locking does not happen automatically and must be specified at run

A True
B False

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

A False
B True

You cannot publish your own modules on the Terraform Registry.

A False
B True

Select all features which are exclusive to Terraform Enterprise. (select three)

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

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 Variables
B Functions
C Expressions
D Local Values

You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a _____________.

A Partial configuration
B Default configuration
C Incomplete configuration
D Changing configuration

The terraform state command can be used to ____

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

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 Backends
C Workspaces
D Environment Variables

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

Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 …) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region. What should you do? Your provider setting is as below -

A Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.
B No way to enable this via a single script . Write 2 different scripts with different default providers in the 2 scripts , one for us-east , another for us-west.
C Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).
D Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.

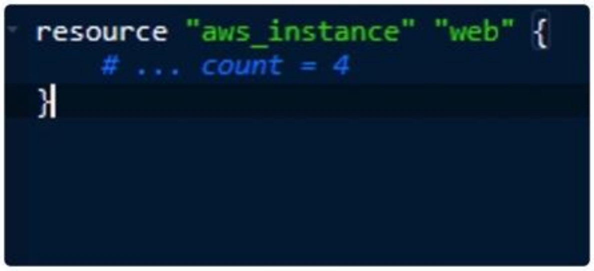
Which one is the right way to import a local module names consul

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

Your team lead does not trust the junior terraform engineers who now have access to the git repo . So , he wants you to have some sort of a checking layer , whereby , you can ensure that the juniors will not create any non-compliant resources that might lead to a security audit failure in future. What can you do to efficiently enforce this ?

A Use Terraform OSS Sentinel Lite version , which will save cost , since there is no charge for OSS , but it can still check for most non-compliant rules using Policy-As-Code.
B Create a design /security document (in PDF) and share to the team , and ask them to always follow that document , and never deviate from it.
C Create a git master branch , and implement PR . Every change needs to be reviewed by you , before being merged to the master branch.
D Since your team is using Hashicorp Terraform Enterprise Edition , enable Sentinel , and write Policy-As-Code rules that will check for non-compliant resource provisioning , and prevent/report them.

Given the below resource configuration. What does the terraform resource address aws_instance.web refer to?



```
resource "aws_instance" "web" {
    # ... count = 4
}
```

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

Named workspaces are not a suitable isolation mechanism for strong separation between staging and production?

A True
B False

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

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

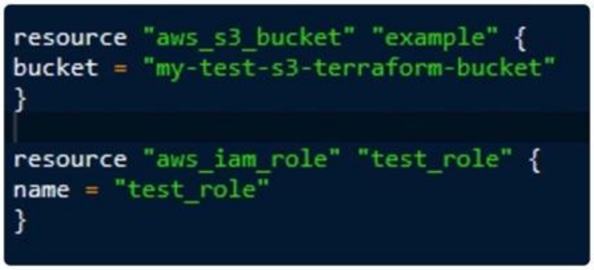
You have written a terraform IaC script which was working till yesterday , but is giving some vague error from today , which you are unable to understand . You want more detailed logs that could potentially help you troubleshoot the issue , and understand the root cause. What can you do to enable this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

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

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

Due to the way that the application code iswritten , the s3 bucket must be created before the test role is created , otherwise there will be a problem. How can you ensure that?



```
resource "aws_s3_bucket" "example" {
bucket = "my-test-s3-terraform-bucket"
}

resource "aws_iam_role" "test_role" {
name = "test_role"
}
```

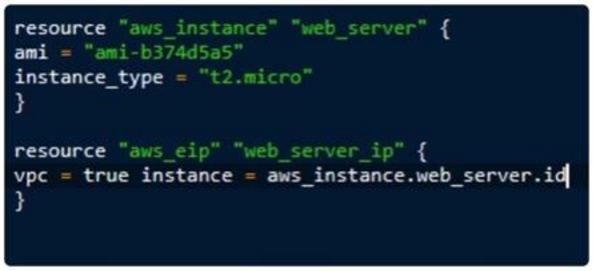
A Add explicit dependency using depends on . This will ensure the correct order of resource creation.
B Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.
C This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.
D This will already be taken care of by terraform native implicit dependency. Nothing else needs to be done from your end.

Which of the following Terraform files should be ignored by Git when committing code to a repository? (select two)

A terraform.tfstate

B variables.tf
C output.tf
D terraform.tfvars

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



```
resource "aws_instance" "web_server" {
ami = "ami-b374d5a5"
instance_type = "t2.micro"
}

resource "aws_eip" "web_server_ip" {
vpc = true instance = aws_instance.web_server.id
}
```

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

Which of the following actions are performed during a terraform init?

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

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

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

Multiple providers can be declared within a single Terraform configuration file.

A False
B True

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 can manage cross-cloud dependencies
D Terraform simplifies management and orchestration, helping operators build large-scale, multi-cloud infrastructure

True or False: A list(...) contain a number of values of the same type while an object(...) can contain a number of values of different types.

A True
B False

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

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

Using multi-cloud and provider-agnostic tools provides which of the following benefits?

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

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

In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

A 1

B 2
C 5
D 3

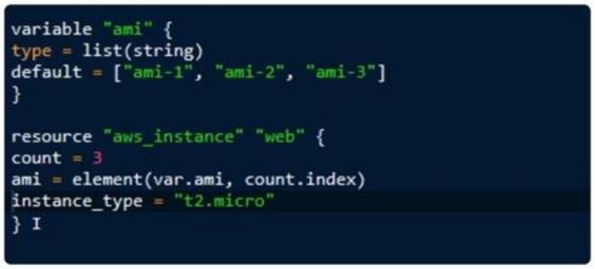
What is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault?

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

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

Refer to the following Terraform code variable If count.index is set to 2, which of the following values will be used?



```
variable "ami" {
type = list(string)
default = ["ami-1", "ami-2", "ami-3"]
}

resource "aws_instance" "web" {
count = 3
ami = element(var.ami, count.index)
instance_type = "t2.micro"
} I
```

A ami
B ami-3
C ami-1
D ami-2

Terraform-specific settings and behaviors are declared in which configuration block type?

A terraform
B resource
C provider
D data
After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?

A INFO
B WARN
C ERROR

D TRACE
E DEBUG

You can migrate the Terraform backend but only if there are no resources currently being managed.

A True
B False

Which of the following is considered a Terraform plugin?

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

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

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

What are some of the features of Terraform state?

A Determining the correct order to destroy resources
B Increased performance
C Mapping configuration to real-world resources
D Inspection of cloud resources

Which configuration file formats are supported by Terraform? (Select all that apply)

A JSON
B Node
C YAML
D HCL
E Go

You've 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 block

B terraform import
C dynamic backend
D splat expression

Which of the following best describes a Terraform provider?

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

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

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

Provisioners should only be used as a last resort.

A False
B True

What Terraform feature is most applicable for managing small differences between different environments, for example development and production?

A Environment Variables
B Backends
C Workspaces
D Repositories

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 destroyed and recreated
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 updated in place

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

A Authentication using MFA
B An API token
C Authentication using MFA

D A TOTP token

What happens when a terraform apply command is executed?

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

In the example below, where is the value of the DNS record's IP address originating from?

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