Attempt 1

All questions

Top of Form

Question 1: **Correct**

**Select all names that are NOT valid variable identifers(select 3)**

* ​

version

**(Correct)**

* ​

variable

* ​

source

**(Correct)**

* ​

lifecycle

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 2: **Correct**

**Identifiers can contain (select 3)**

* ​

digits

**(Correct)**

* ​

tilde

* ​

hyphens

**(Correct)**

* ​

underscores

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 3: **Correct**

**Vault Provider is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault**

* ​

TRUE

**(Correct)**

* ​

FALSE

**Explanation**

The Vault provider helps Transfrom to read, write, and configure Hashicorp Vault.

Bottom of Form

Top of Form

Question 4: **Correct**

**Which command can be used to launch the Interactive console for Terraform interpolations?**

* ​

terraform terminal

* ​

terraform cmd

* ​

terraform cli

* ​

terraform console

**(Correct)**

**Explanation**

The terraform console command provides an interactive console for evaluating expressions.

Bottom of Form

Top of Form

Question 5: **Correct**

**\_\_\_\_\_\_\_\_ command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure.**

* ​

terraform refresh

**(Correct)**

* ​

terraform reset

* ​

terraform update

* ​

terraform reconcile

**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.

Bottom of Form

Top of Form

Question 6: **Correct**

**Terraform has special handling of any configuration file whose name ends in \_\_\_\_\_\_\_(select 3)**

* ​

\_override.tf

**(Correct)**

* ​

override.tf

**(Correct)**

* ​

\_override.tf.txt

* ​

\_override.tf.json

**(Correct)**

**Explanation**

Terraform has special handling of any configuration file whose name ends in \_override.tf or \_override.tf.json. This special handling also applies to a file named literally override.tf or override.tf.json. https://www.terraform.io/docs/configuration/override.html

Bottom of Form

Top of Form

Question 7: **Correct**

**When using local state, state is stored in \_\_\_\_\_\_ files.**

* ​

plain-text XML

* ​

plain-text LOG

* ​

plain-text TXT

* ​

plain-text JSON

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 8: **Correct**

**Which of the following are Terraform string functions? (select 3)**

* ​

replace

**(Correct)**

* ​

tostring

* ​

join

**(Correct)**

* ​

format

**(Correct)**

**Explanation**

tostring is not a string function, it is a type conversion function.

Bottom of Form

Top of Form

Question 9: **Incorrect**

**Which configuration block type is used to declare Terraform-specific settings and behaviors?**

* ​

terraform

**(Correct)**

* ​

dev

* ​

data

* ​

provider

**(Incorrect)**

**Explanation**

The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.

Bottom of Form

Top of Form

Question 10: **Correct**

**In Terraform Enterprise, a workspace can be mapped only to one VCS repos**

* ​

TRUE

**(Correct)**

* ​

FALSE

**Explanation**

A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo, if needed.

Bottom of Form

Top of Form

Question 11: **Correct**

**User tokens do not inherit permissions from the user they are associated with.**

* ​

FALSE

**(Correct)**

* ​

TRUE

**Explanation**

User tokens are the most flexible token type because they inherit permissions from the user they are associated with.

Bottom of Form

Top of Form

Question 12: **Correct**

**Which of the following are supported type keywords in Terraform? (select 3)**

* ​

integer

* ​

bool

**(Correct)**

* ​

string

**(Correct)**

* ​

number

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 13: **Correct**

**When using local-state, the state file is NOT stored as plain-text**

* ​

FALSE

**(Correct)**

* ​

TRUE

**Explanation**

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

Bottom of Form

Top of Form

Question 14: **Incorrect**

**required\_versions is the flag used within a Terraform configuration block to identify the specific version of a provider. ​**

* ​

FALSE

**(Correct)**

* ​

TRUE

**(Incorrect)**

**Explanation**

required\_providers is the flag used within a Terraform configuration block to identify the specific version of a provider.

Bottom of Form

Top of Form

Question 15: **Incorrect**

**Terraform helps in \_\_\_\_\_\_ (select 3)**

* ​

configuration management

**(Incorrect)**

* ​

initializing resources

**(Correct)**

* ​

higher-level abstraction of the datacenter

**(Correct)**

* ​

Bootstrapping

**(Correct)**

**Explanation**

Terraform is not a configuration management tool, and it allows existing tooling to focus on their strengths: bootstrapping and initializing resources. https://www.terraform.io/intro/vs/chef-puppet.html

Bottom of Form

Top of Form

Question 16: **Incorrect**

**The local backend is how Terraform connects to public cloud services such as AWS, GCP.**

* ​

TRUE

**(Incorrect)**

* ​

FALSE

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 17: **Correct**

**HCL stands for \_\_\_\_\_\_\_**

* ​

HashiCorp Configuration Language.

**(Correct)**

* ​

HashiCorp Confirmation Language.

* ​

HashiCorp Creation Language.

* ​

HashiCorp Configure Language.

**Explanation**

HashiCorp Configuration Language.

Bottom of Form

Top of Form

Question 18: **Incorrect**

**What are the some of the log levels of TF\_LOG? (select 3)**

* ​

DEBUG

**(Correct)**

* ​

TRACE

**(Correct)**

* ​

INFO

**(Correct)**

* ​

ACCESS

**(Incorrect)**

**Explanation**

You can set TF\_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs.

Bottom of Form

Top of Form

Question 19: **Incorrect**

**Terraform store its state file in a shared directory**

* ​

FALSE

**(Correct)**

* ​

TRUE

**(Incorrect)**

**Explanation**

Terraform store its state file in the current working directory.

Bottom of Form

Top of Form

Question 20: **Incorrect**

**What is the recommended number of spaces between each nesting level?**

* ​

1

* ​

3

**(Incorrect)**

* ​

4

* ​

2

**(Correct)**

**Explanation**

HashiCorp style conventions state that you should use 2 spaces between each nesting level to improve the readability

Bottom of Form

Top of Form

Question 21: **Correct**

**Which of the following is a valid Terraform string function?**

* ​

delegate

* ​

format

**(Correct)**

* ​

edit

* ​

nullify

Bottom of Form

Top of Form

Question 22: **Correct**

**Which of the following Terraform files should be added to .gitignore before committing code to a Github Repository? (select 2)**

* ​

terraform.tf

* ​

terraform.state

* ​

terraform.tfvars

**(Correct)**

* ​

terraform.tfstate

**(Correct)**

**Explanation**

The terraform.tfstate file contains the terraform state of a specific environment and doesn't need to be preserved in a repo. The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others. Hence these two files should be ignored.

Bottom of Form

Top of Form

Question 23: **Correct**

**What are some of the features of Terraform state? (select 3)**

* ​

Cloud Resource Inspection

* ​

Metadata Storage

**(Correct)**

* ​

Performance Improvement

**(Correct)**

* ​

Mapping configuration to resources in the real world

**(Correct)**

**Explanation**

https://www.terraform.io/docs/state/purpose.html

Bottom of Form

Top of Form

Question 24: **Correct**

**The \_\_\_\_\_\_ command is used to rewrite Terraform configuration files to a canonical format and style**

* ​

terraform stylize

* ​

terraform format

* ​

terraform style

* ​

terraform fmt

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 25: **Correct**

**\_\_\_\_\_\_\_ retrieves the value of a single element from a map, given its key.**

* ​

get

* ​

read

* ​

pull

* ​

lookup

**(Correct)**

**Explanation**

lookup retrieves the value of a single element from a map, given its key. If the given key does not exist, a the given default value is returned instead.

Bottom of Form

Top of Form

Question 26: **Correct**

**Which is the command used to switch to the dev workspace?**

* ​

terraform workspace switch dev

* ​

terraform dev workspace switch

* ​

terraform switch dev

* ​

terraform workspace select dev

**(Correct)**

**Explanation**

The terraform workspace select  command is used to choose a different workspace to use for further operations.

Bottom of Form

Top of Form

Question 27: **Correct**

**API and CLI access are managed with \_\_\_\_\_\_\_**

* ​

API tokens

**(Correct)**

* ​

email and password

* ​

TOTP tokens

* ​

username and password

**Explanation**

API and CLI access are managed with API tokens, which can be generated in the Terraform Cloud UI.

Bottom of Form

Top of Form

Question 28: **Correct**

**Which are some of the different collection types available in Terraform(select 3)**

* ​

list(<TYPE>)

**(Correct)**

* ​

set(<TYPE>)

**(Correct)**

* ​

map(<TYPE>)

**(Correct)**

* ​

class(<TYPE>)

**Explanation**

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

Bottom of Form

Top of Form

Question 29: **Correct**

**Which of the following connection types are supported by the remote-exec provisioner? (select 2)**

* ​

ssh

**(Correct)**

* ​

tdp

* ​

winrm

**(Correct)**

* ​

vinrm

**Explanation**

The remote-exec provisioner supports both ssh and winrm type connections.

Bottom of Form

Top of Form

Question 30: **Correct**

**\_\_\_\_\_\_ enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.**

* ​

module registry

* ​

workspaces

* ​

sentinel

**(Correct)**

* ​

functions

**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.

Bottom of Form

Top of Form

Question 31: **Incorrect**

**Terraform Cloud's organizational and access control model is based on two units: users and organizations.**

* ​

FALSE

**(Correct)**

* ​

TRUE

**(Incorrect)**

**Explanation**

Terraform Cloud's organizational and access control model is based on three units: users, teams, and organizations.

Bottom of Form

Top of Form

Question 32: **Incorrect**

**terraform apply command is used to apply the changes required to reach the desired state of the configuration.**

* ​

FALSE

**(Incorrect)**

* ​

TRUE

**(Correct)**

Bottom of Form

Top of Form

Question 33: **Correct**

**\_\_\_\_\_\_\_\_\_ is used within a Terraform configuration block to identify the specific version of a provider**

* ​

required\_versions

* ​

required-providers

* ​

required-versions

* ​

required\_providers

**(Correct)**

Bottom of Form

Top of Form

Question 34: **Correct**

**For local state, Terraform stores the workspace states in a directory called \_\_\_\_\_\_\_\_\_\_**

* ​

terraform.tfstate.d

**(Correct)**

* ​

terraform.d

* ​

tfstate.d

* ​

terraform.tfstate.directory

**Explanation**

For local state, Terraform stores the workspace states in a directory called terraform.tfstate.d.

Bottom of Form

Top of Form

Question 35: **Correct**

**The command terraform fmt updates the font of the configuration file to the official font supported by HashiCorp**

* ​

TRUE

* ​

FALSE

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 36: **Correct**

**A dynamic block is used to iterate over a list of required TCP ports to add to the new security group.**

* ​

TRUE

**(Correct)**

* ​

FALSE

**Explanation**

A dynamic block 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.

Bottom of Form

Top of Form

Question 37: **Correct**

**How many resources can Terraform provision concurrently?**

* ​

15

* ​

12

* ​

25

* ​

10

**(Correct)**

**Explanation**

Terraform can limit the number of concurrent operations as Terraform walks the graph using the -parallelism=n argument. The default value for this setting is 10. This setting might be helpful if you're running into API rate limits.

Bottom of Form

Top of Form

Question 38: **Incorrect**

**Workspace names need to be 90 characters or less and can only include letters, numbers, hyphens, and underscore.**

* ​

TRUE

**(Correct)**

* ​

FALSE

**(Incorrect)**

Bottom of Form

Top of Form

Question 39: **Correct**

**In order to make a Terraform configuration file dynamic and/or reusable, static values should be converted to input parameters**

* ​

FALSE

* ​

TRUE

**(Correct)**

**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.

Bottom of Form

Top of Form

Question 40: **Incorrect**

**Workspaces allow Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code?**

* ​

FALSE

**(Correct)**

* ​

TRUE

**(Incorrect)**

**Explanation**

Sentinel allows Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code.

Bottom of Form

Top of Form

Question 41: **Incorrect**

**Consul Enterprise uses Sentinel to augment the built-in ACL system to provide advanced policy enforcement.**

* ​

FALSE

**(Correct)**

* ​

TRUE

**(Incorrect)**

**Explanation**

Nomad Enterprise uses Sentinel to augment the built-in ACL system to provide advanced policy enforcement.

Bottom of Form

Top of Form

Question 42: **Correct**

**Which of the following a Terraform providers? (select 3)**

* ​

AWS

**(Correct)**

* ​

Bubble

* ​

Azure

**(Correct)**

* ​

Alibaba Cloud

**(Correct)**

**Explanation**

https://www.terraform.io/docs/providers/index.html

Bottom of Form

Top of Form

Question 43: **Correct**

**Terraform Provider is a Terraform plugin**

* ​

TRUE

**(Correct)**

* ​

FALSE

**Explanation**

Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins.

Bottom of Form

Top of Form

Question 44: **Correct**

**The \_\_\_\_\_\_ command is used to create an execution plan.**

* ​

terraform plan

**(Correct)**

* ​

terraform apply

* ​

terraform create

* ​

terraform init

**Explanation**

The terraform plan command is used to create an execution plan.

Bottom of Form

Top of Form

Question 45: **Incorrect**

**The key features of Terraform are \_\_\_\_\_(select 3)**

* ​

Infrastructure as Code

**(Correct)**

* ​

Change Automation

**(Correct)**

* ​

Advanced Pipeline Streaming

**(Incorrect)**

* ​

Resource Graph

**(Correct)**

**Explanation**

https://www.terraform.io/intro/index.html

Bottom of Form

Top of Form

Question 46: **Correct**

**Consul is a tool for \_\_\_\_\_\_ (select 3)**

* ​

configuration

**(Correct)**

* ​

service discovery

**(Correct)**

* ​

orchestration

**(Correct)**

* ​

resource management

**Explanation**

Consul is a tool for service discovery, configuration and orchestration. https://www.terraform.io/intro/examples/consul.html

Bottom of Form

Top of Form

Question 47: **Incorrect**

**A Terraform Enterprise install that is provisioned on a network that does not have Internet access is generally known as \_\_\_\_\_\_\_**

* ​

an air-gapped install

**(Correct)**

* ​

a public install

* ​

a private install

**(Incorrect)**

* ​

a connected install

**Explanation**

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

Bottom of Form

Top of Form

Question 48: **Incorrect**

**What are some of the use cases of Terraform? (select 3)**

* ​

Single-Cloud Deployment

**(Incorrect)**

* ​

Software Demos

**(Correct)**

* ​

Heroku App Setup

**(Correct)**

* ​

Multi-Tier Applications

**(Correct)**

**Explanation**

https://www.terraform.io/intro/use-cases.html

Bottom of Form

Top of Form

Question 49: **Incorrect**

**Which are some of the optional arguements avaialable in output blocks(select 3)**

* ​

depends\_on

**(Correct)**

* ​

description

**(Correct)**

* ​

sensitive

**(Correct)**

* ​

format

**(Incorrect)**

**Explanation**

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

Bottom of Form

Top of Form

Question 50: **Correct**

**Terraform Cloud can be managed from the CLI but requires an API Token**

* ​

FALSE

* ​

TRUE

**(Correct)**

**Explanation**

API and CLI access are managed with API tokens, which can be generated in the Terraform Cloud UI.

Bottom of Form

Top of Form

Question 51: **Correct**

**The terraform taint Terraform command will force a marked resource to be destroyed and recreated on the next apply**

* ​

FALSE

* ​

TRUE

**(Correct)**

**Explanation**

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.

Bottom of Form

Top of Form

Question 52: **Incorrect**

**A terraform list is a sequence of values identified by consecutive whole numbers starting with \_\_\_\_\_**

* ​

zero

**(Correct)**

* ​

one

**(Incorrect)**

* ​

negative one

* ​

null

**Explanation**

A terraform list is a sequence of values identified by consecutive whole numbers starting with zero.

Bottom of Form

Top of Form

Question 53: **Correct**

**tostring is a \_\_\_\_\_\_ Function**

* ​

String

* ​

Integer

* ​

Type conversion

**(Correct)**

* ​

Text

**Explanation**

tostring is a type conversion function. tostring converts its argument to a string value.

Bottom of Form

Top of Form

Question 54: **Correct**

**Terraform Cloud supports three distinct types of API tokens with varying levels of access: user, team, and organization.**

* ​

TRUE

**(Correct)**

* ​

FALSE

Bottom of Form

Top of Form

Question 55: **Correct**

**If you are an organization owner you can require all users within your organization to use two-factor authentication.**

* ​

FALSE

* ​

TRUE

**(Correct)**

Bottom of Form

Top of Form

Question 56: **Correct**

**Which of these are valid Numeric Functions in Terraform (select 2)**

* ​

pow

**(Correct)**

* ​

abs

**(Correct)**

* ​

root

* ​

format

**Explanation**

abs and pow are some valid Numeric functions. https://www.terraform.io/docs/configuration/functions.html

Bottom of Form

Top of Form

Question 57: **Correct**

**Terraform is generally written using?**

* ​

XML

* ​

Typescript

* ​

YAML

* ​

HCL

**(Correct)**

**Explanation**

Terraform is generally written using the HashiCorp Configuration Language (HCL)

Bottom of Form

Top of Form

Question 58: **Correct**

**Which of the following Terraform features are only available in the Enterprise edition? (select 3)**

* ​

Clustering

**(Correct)**

* ​

Audit Logs

**(Correct)**

* ​

SAML/SSO

**(Correct)**

* ​

Private Module Registry

**Explanation**

To see what specific features are part of Terraform Cloud and Terraform Enterprise, Go to https://www.hashicorp.com/products/terraform/pricing/

Bottom of Form

Top of Form

Question 59: **Incorrect**

**Select all statements that are TRUE for Terraform expressions(select 3)**

* ​

Bools are represented by the unquoted symbols true and false.

**(Correct)**

* ​

Lists/tuples are represented by a pair of curly braces containing a series of <KEY> = <VALUE> pairs

**(Incorrect)**

* ​

The null value is represented by the unquoted symbol null

**(Correct)**

* ​

Numbers are represented by unquoted sequences of digits with or without a decimal point

**(Correct)**

**Explanation**

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

Bottom of Form

Top of Form

Question 60: **Correct**

**What are some of the different syntaxes for comments? (select 3)**

* ​

#

**(Correct)**

* ​

/\* and \*/

**(Correct)**

* ​

//

**(Correct)**

* ​

\*/ and /\*

**Explanation**

Terraform supports the #, //, and /\*..\*/ for commenting Terraform configuration files. https://www.terraform.io/docs/configuration/syntax.html#comments

Bottom of Form