1. Which one among these configuration formats is supported by terraform? (Choose 2)- HCL and JSON
2. Which among these offerings are unique to Terraform Enterprise and are not available in Terraform Open Source as well as Terraform Cloud (Free and Team & Governance). Select 3.

Ans: auto logging, clustering functionality and private network connectivity, SSO/SAML

1. It is recommended that terraform init should not be run multiple times since the command will download all the plugins from the cloud provider every time the command is run. -false
2. After the TF\_LOG\_PATH is set by the administrator, is it also necessary to set the TF\_LOG for the logging to be enabled? -yes , it is required
3. Does Terraform Workspace in the open-source version provide similar functionality when compared to Terraform Enterprise and the Cloud version? – No /False
4. Which terraform command can rewrite Terraform configuration files to a canonical format? – terraform fmt
5. There has been an open issue related to Terraform. You have figured out a solution and would like to add a fix to the terraform project. Which language you will need to write the fix? - GO
6. Which among the following definition best describes default local backend in Terraform OSS?-

stores state file on local system, locks the state using system APIs, and performs operations locally

1. Which among the following is the default backend for Terraform OSS?- local
2. Which of the following represents a feature of Terraform Cloud that is NOT free to customers? – Roles and Team management
3. Terraform works well in Windows but a Windows server is required?- NO/False
4. Which command launches an interactive console?- terraform console
5. Can we delete the default workspace in Terraform?- no
6. Terraform backend can only be migrated if there is no resource being managed by terraform? \_False
7. Terraform parallelism allows us to limit the number of concurrent operation as Terraform. By default, how many resources will Terraform provision concurrently? -10
8. Do we have to rebuild the state file locally if a remote backend is removed from the configuration? -NO

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](https://www.terraform.io/docs/backends/init.html). 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.

1. Which of the following operating system is supported by Terraform Enterprise? – UBUNTU, oracle linux, red hat enterprise linux, CentOs, Amazon Linux, Debian

Not support -UNIX

1. terraform fmt will scan all the sub-directories of the current working directory were the command was run – False

By default, fmt scans the current directory for configuration files. If the dir argument is provided then it will scan that given directory instead

1. 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.- NO, As of now, there is no mechanism to protect secrets in Terraform.
2. Select all Operating Systems that Terraform is available for. (select five) – LINUX, MACOS, Solaris, FreeBSd, Windows.

Not available for Unix

1. What are the ways in which certain data like secrets can be omitted so they are not shared with others? –

* Command line key/value pairs
* Use the -backend-config =PATH to specify separate config file
* Interactively on command line
* **Give an example of passing partial configuration with Command-line Key/Value pairs?**
* terraform init \  
   -backend-config="address=demo.consul.io" \  
   -backend-config="path=example\_app/terraform\_state" \  
   -backend-config="scheme=https"

1. The name of a variable can be any valid identifier except the following: source, version, providers, count, for\_each, lifecycle, depends\_on, locals.
2. What is the easiest and recommended way for Terraform to read secrets from HashiCorp Vault? – Vault Provider
3. Which is not an example of infrastructure as a code?

An interactive web catalog

1. Not a string function : toString, Slice
2. When multiple arguments with single-line values appear on consecutive lines at the same nesting level, HashiCorp recommends that you: Align the = sign

27.

Select all features which are exclusive to Terraform Enterprise and Terraform Cloud for Business (select three).

Single Sign On (SSO), Audit Logging, Self-Hosted Agent

1. 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, assuming you are following the standard workflow of write, plan, and apply? - in Plan
2. In Terraform Enterprise, a workspace can be mapped to how many VCS repos? – only 1
3. What are some of the features of Terraform state? (select three)- increase performance, deterime ordering for creating and destroying resources, mapping conigurations to real world infra
4. Starting in Terraform v0.12, the Terraform language now has built-in syntax for creating lists using the [ and ] delimiters, replacing and deprecating the list () function. \_TRUE
5. What are the core Terraform workflow steps to use infrastructure as code? – Write, Plan, Apply

The prefix -/+ means that Terraform will destroy and recreate the resource, rather than updating it in-place. Some attributes and resources can be updated in-place and are shown with the ~ prefix.

1. 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 one)- Clustering
2. **Only constants are allowed inside the terraform block. Is this correct?- Yes**
3. ***What is the CLI configuration File?***
4. The CLI configuration file configures per-user settings for CLI behaviors, which apply across all Terraform working directories.It is named either ***.terraformrc*** or ***terraform.rc***
5. ***Where is the location of the CLI configuration File?***
6. On Windows, the file must be named named **terraform.rc** and placed in the relevant user's **%APPDATA%** directory.On all other systems, the file must be named **.terraformrc** (note the leading period) and placed directly in the home directory of the relevant user.The location of the Terraform CLI configuration file can also be specified using the **TF\_CLI\_CONFIG\_FILE** [environment variable](https://www.terraform.io/docs/commands/environment-variables.html).

38. **What is Provider Plugin Cache?**

By default, terraform init downloads plugins into a subdirectory of the working directory so that each working directory is self-contained. As a consequence, if you have multiple configurations that use the same provider then a separate copy of its plugin will be downloaded for each configuration.Given that provider plugins can be quite large (on the order of hundreds of megabytes), this default behavior can be inconvenient for those with slow or metered Internet connections. Therefore Terraform optionally allows the use of a local directory as a shared plugin cache, which then allows each distinct plugin binary to be downloaded only once.

**39. How do you enable Provider Plugin Cache?**

To enable the plugin cache, use the plugin\_cache\_dir setting in [the CLI configuration file](https://www.terraform.io/docs/commands/cli-config.html).

**plugin\_cache\_dir** **=** "$HOME/.terraform.d/plugin-cache

"Alternatively, the **TF\_PLUGIN\_CACHE\_DIR** environment variable can be used to enable caching or to override an existing cache directory within a particular shell session:

40. **When you are doing initialization with terraform init, you want to skip backend initialization. What should you do?**

terraform init -backend=false

41. **When you are doing initialization with terraform init, you want to skip child module installation. What should you do?**

terraform init -get=false

42. **When you are doing initialization with terraform init, you want to skip plugin installation. What should you do?**

terraform init -get-plugins=false

43. **You are applying the infrastructure with the command apply and you don’t want to do interactive approval. Which flag should you use?**

terraform apply -auto-approve

44. **How do you preview the behavior of the command terraform destroy?**

terraform plan -destroy

45. **How do you save the execution plan?**

terraform plan -out=tfplan

46. **You have started writing terraform configuration and you are using some sample configuration as a basis. How do you copy the example configuration into your working directory?**

terraform init -from-module=MODULE-SOURCE

47. **What is the flag you should use with the terraform plan to get detailed on the exit codes?**

terraform plan -detailed-exitcode

\* 0 = Succeeded with empty diff (no changes)  
\* 1 = Error  
\* 2 = Succeeded with non-empty diff (changes present)

48. **If you want to switch from using remote backend to local backend. What should you do?**

If you want to move back to local state, you can remove the backend configuration block from your configuration and run **terraform init again.** Terraform will once again ask if you want to migrate your state back to local.

49. **Does environment variables support List and map types?**

NoEnvironment variables can only populate string-type variables.

50. **How do you provision infrastructure in a staging environment or a production environment using the same Terraform configuration?**

You can use different varible files with the same configuration// Example// For development  
terraform apply -var-file="dev.tfvars"// For test  
terraform apply -var-file="test.tfvars"