**1. What is Terraform?**

- Tool for infrastructure provisioning

- It allows automate and manage your infrastructure and platform of services

- Open source and declarative

- With Terraform we can create VPC, username/permission on AWS, Azure and other cloud platforms, install docker…

- Managing Infrastructure

- Replicating Infrastructure

**Note:** Ansible is infrastructure configuration tool but Terraform infrastructure provisioning tool

**2. What are the most useful Terraform commands?**

* terraform init - initializes the current directory
* terraform refresh - refreshes the state file
* terraform output - views Terraform outputs
* terraform apply - applies the Terraform code and builds stuff
* terraform destroy - destroys what has been built by Terraform
* terraform graph - creates a DOT-formatted graph
* terraform plan - a dry run to see what Terraform will do

**3. What are Modules in Terraform?**

Modules are collections of .tf files containing resources, input variables, and outputs, which exist outside the root folder of your configuration.

**4. What is the Terraform Plugins?**

Terraform Plugins are executable binaries written in Go programming language. Plugins are basically the providers and provisioners in Terraform configurations.

**5. What is Execution Plans in Terraform?**

Terraform has a 'planning' step where it generates an execution plan. The execution plan shows what Terraform will do when the call apply. This will avoid any surprises when Terraform manipulates infrastructure.

**6. Define Terraform init?**

The Terraform init is a control that is used to initialize an operational index containing Terraform pattern files. This is the first authority that should be sprint after writing a new Terraform design or cloning an obtainable one from account control. It is safe to lope this control multiple times.

**7. If different teams are working on the same configuration. How do you make files to have syntactically valid and internally consistent?**

terraform validate

This command will check and report errors within modules, attribute names, and value types. Validate your configuration. If your configuration is valid, Terraform will return a success message.

**8. How do you inspect the current state of the infrastructure applied?**

terraform show

When you applied your configuration, Terraform wrote data into a file called terraform.tfstate. This file now contains the IDs and properties of the resources Terraform created so that it can manage or destroy those resources going forward.

**9. Explain State File Locking?**

State file locking is Terraform mechanism in which operations on a specific state file are blocked to avoid conflicts between multiple users performing the same process. When one user releases the lock, then only the other one can operate on that state. This helps in preventing state file corruption. This is a backend operation.

Note: Terraform keeps track of all the resources created in a state file

**10. What do you understand by a Tainted Resource?**

A tainted resource is a resource that is forced to be destroyed and recreated on the next apply command. When a resource is marked as tainted, the state files are updated, but nothing changes on infrastructure. The terraform plan out shows that help will get destroyed and recreated. The changes get implemented when the next apply happens.