TERRAFORM INTERVIEW QUESTIONS

1. Why we required IaC?

   To meet customer demands and save costs, application teams are architecting their applications to support a much higher level of elasticity, supporting technology like containers and public cloud resources. These resources may only live for a matter of hours; therefore, the traditional method of raising a ticket to request resources is no longer a viable option. So, we can use IaC to meet the above requirements.

2. What do you mean by IaC or what is IaC?

   Infrastructure as Code (IAC) is the managing and provisioning of infrastructure through code instead of through manual process.

3. What is Terraform?

   Terraform is basically a tool which is used to create and maintain the infrastructure through code by its own configuration language HCL (Hashicorp Configuration Language) instead of doing manual process. It was written in GoLang and developed by Mitchell Hashimoto in the year 2014. It achieves its work by maintaining the actual state equal to desired state.

4. Explain different terraform commands like

   a. **Init:** The terraform init command is used to initialize a working directory containing terraform configuration files. It is safe to run this command multiple times. This command will never delete your existing configuration or state. During init, the root configuration directory is consulted for backed configuration and the chosen backend is initialized using the given configuration settings

   b. **Plan:** The terraform plan command is used to create an execution plan. Terraform performs a refresh, unless explicitly disable and then determines what actions are necessary to achieve the desired state specified in the configuration files.

   c. **Validate:** The terraform validate command validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs etc. validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus useful in general verification of reusable modules including correctness of attribute names and value types.

d. **Apply:** The terraform apply command is used to apply the changes required to reach the desired state of the configuration or the pre-determined set of actions generated by a terraform plan execution plan.

e. **Destroy:** The terraform destroy command is used to destroy the terraform-managed infrastructure.

5. All the conceptual meanings w.r.t terraform –
   a. Terraform variables
   b. Terraform input types
   c. Terraform dependencies
   d. Terraform arguments
   e. Terraform attributes
   f. Terraform functions
   g. Terraform modules
   h. Terraform data source
   i. Terraform taint
   j. Terraform untaint
   k. Terraform format
   l. Terraform provisioning
   m. Terraform state (state_files)
   n. Terraform workspace
   o. Terraform execution process/ work flow
   p. Terraform state file locking
   q. Terraform core
   r. Terraform remote backend
   s. Terraform multiple providers

6. What is the difference between Terraform and Ansible?

   Ansible, chef, puppet are configuration management tools which means they are primarily designed to install and manage software on existing servers.

Terraform and cloud-formation are IAC tools which are designed to provision servers and infrastructure themselves. You can use IAC and configuration management tools altogether.
For e.g.: you can use terraform to create a new EC2 instance on AWS, terraform can then call ansible to install and configure software and applications in the EC2 instances.

7. What is the difference between Terraform and CloudFormation?

8. What is the difference between IaC and Configuration Management?

   Configuration management tools which mean they are primarily designed to install and manage software on existing servers.
   IAC tools which are designed to provision servers and infrastructure themselves.
   You can use IAC and configuration management tools altogether.
   For e.g.: you can use terraform to create a new EC2 instance on AWS, terraform can then call ansible to install and configure software and applications in the EC2 instances.

9. What are the major competitors of Terraform?

   Azure Management Tools.
   Morpheus.
   CloudHealth.
   Turbonomic.
   CloudBolt.
   Apptio Cloudability
   Ansible
   Kubernetes
   AWS Cloud Formation

10. What are the main features of Terraform and use cases?

    Following are the key features of Terraform:
    Infrastructure as Code: Terraform's high-level configuration language is used to describe your infrastructure in declarative configuration files that are human readable. You may now generate a blueprint that you can edit, share, and reuse.

    Execution Strategies: Before making any infrastructure modifications, Terraform develops an execution plan that describes what it will do and asks for your agreement. Before Terraform produces, upgrades, or destroys infrastructure, you can evaluate the changes.

Graph of Resources: Terraform develops or alters non-dependent resources while simultaneously building a resource graph. This allows Terraform to construct resources as quickly as possible while also providing you with more information about your infrastructure.

Automation of Change: Terraform can automate the application of complex changesets to your infrastructure with little to no human intervention. Terraform identifies what happened when you update configuration files and provides incremental execution plans that take dependencies into account.

11. What is Terragrunt?

Terragrunt is a lightweight wrapper that adds extra features for maintaining DRY configurations, dealing with many Terraform modules, and managing remote state.

12. Why Terraform is referred as one of the DevOps tools?

- ✓ Terraform allows you to specify infrastructure in config/code, making it simple to rebuild, alter, and track infrastructure changes. Terraform is a high-level infrastructure description. While there are a few alternatives, they are all centred on a single cloud provider.
- ✓ Terraform is the only powerful solution that is totally platform-neutral and supports different services.
- ✓ Terraform allows you to implement a variety of coding concepts, such as putting your code under version control, writing automated tests, and so on.
- ✓ Terraform is the best tool for infrastructure management since many other solutions suffer from an impedance mismatch when attempting to use an API meant for configuring management to govern an infrastructure environment. Instead, Terraform is a perfect match for what you want to do because the API is built around how you think about infrastructure.
- ✓ Terraform has a thriving community and is open source, so it's attracting a sizable following. Many people already use it, making it easy to discover individuals who know how to use it, as well as plugins, extensions, and expert assistance. Terraform is also evolving at a much faster rate as a result of this. They have a lot of releases.

- ✓ Terraform's speed and efficiency are unrivalled. Terraform's plan command, for example, allows you to see what changes you're about to make before you do them.
- ✓ Terraform and its code reuse feature makes most modifications faster than similar tools like CloudFormation.

13. How to handle roll backs in Terraform?

    In our Version Control System, we need to recommit the previous code version to make it the new and current one. This would start the terraform run command, which would execute the old code. Because terraform is more declarative, we will make sure that everything in the code reverts to its previous state. If the state file becomes corrupted, we would use Terraform Enterprise's State Rollback feature to restore the previous state.

14. How can you prevent duplicate resource errors in Terraform.

    Depending on the situation and the necessity, it can be accomplished in one of three ways. By destroying the resource, the Terraform code will no longer manage it. By removing resources from APIs Importing action will also aid in resource elimination.

15. Be prepared for some HCL syntax of Script for any of the cloud provider you have knowledge like AWS/Azure/GCP.