**Terraform interview questions**

1. What is Terraform init, and how does it work?

Terraform init is a control that allows you to set up an operational index with Terraform pattern files. This control can be repeated several times. It should be the initial command executed after creating a new Terraform design.

2. What is the purpose of Terraform in DevOps?

The HashiCorp Configuration Language, which is comparable to JSON, is used by Terraform (HCL). HCL provides a concise vocabulary that makes establishing and enforcing infrastructure settings across multiple clouds and on-premises data centres simple for DevOps teams.

3. What is "terraform backend"?

A backend may be specified in any Terraform setup, which defines two key things:

Where are surgeries carried out?

Where is the state saved? (In a state file, Terraform maintains track of all the resources generated.)

4. What is a Terraform cloud?

The platform which allows teams to collaborate on Terraform projects on-demand or in reaction to specific circumstances is the Terraform cloud. It is tightly connected with Terraform's processes and data. Terraform modules are shared by a private registry.

5. What version control systems do Terraform support in addition to GitHub?

GitLab EE

GitLab CE

Bucket cloud

6. What is the purpose of Terraform CLI, and what are some common CLI commands?

Terraform's Command-Line Interface (CLI) is used to manage infrastructure and communicate with Terraform state, configuration files, and providers, among other things.

Here are some simple CLI commands to get you started:

Terraform init - Sets up your working directory to run additional commands.

Terraform destroy - Destroys the infrastructure you've already built.

Terraform validate - Check if the configuration is correct using terraform validate.

Terraform apply - This command develops or changes infrastructure.

Terraform plan - Depicts the modifications that the existing arrangement necessitates.

7. Who are Terraform's main competitors?

Terraform's key competitors and alternatives are Azure Management Tools, Morpheus, CloudHealth, Turbonomic, and CloudBolt.

8. What are Terraform modules?

In Terraform, a module is a container containing many resources utilized together. The root module is required for every Terraform and includes the resources listed in the. tf files.

9. How is a duplicate resource error during terraform application ignored?

We can experiment with the following options:

Delete the resources from the cloud provider's API and rebuild them with Terraform.

To stop Terraform from managing specific resources, remove them from the code.

Perform a resource terraform import and delete the code attempting to replicate them.

10. List all of Terraform's version control options.

The following version controls are supported:

Azure DevOps Services.

Azure DevOps Server

Bitbucket Server

Cloud Bitbucket

Gitlab Enterprise Edition (EE) and Enterprise Edition (CE)

Gitlab.com

GitHub Enterprise.

GitHub.com (OAuth)

GitHub.com