**Terraform**

1. Define IAC?

IAC or Infrastructure as Code allows you to build, change, and manage your infrastructure through coding instead of manual processes. The configuration files are created according to your infrastructure specifications and these configurations can be edited and distributed securely within an organization.

1. What are the most useful Terraform commands?

Some of the most useful Terraform commands are:

* 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

1. What is Terraform init?

Terraform init is a control to initialize an operational index that contains Terraform pattern files. This control can be looped multiple times. It is the first command that should be run after writing the new Terraform design.

1. Why is Terraform used for DevOps?

Terraform uses a JSON-like configuration language called the HashiCorp Configuration Language (HCL). HCL has a very simple syntax that makes it easy for DevOps teams to define and enforce infrastructure configurations across multiple clouds and on-premises data centers.

1. What do you understand by terraform backend?

Each Terraform configuration can specify a backend, which defines two main things:

* Where operations are performed
* Where the state is stored (Terraform keeps track of all the resources created in a state file)

1. What are the version controls supported by Terraform besides GitHub?

The supported version controls are:

* Azure DevOps Services
* Azure DevOps Server
* Bitbucket Server
* Bitbucket Cloud
* Gitlab EE and CE
* Gitlab.com
* GitHub Enterprise
* GitHub.com (OAuth)
* GitHub.com

1. Name some major competitors of Terraform?

Some of the top competitors and alternatives to Terraform are Azure Management Tools, Morpheus, CloudHealth, Turbonomic, and CloudBolt.

1. Explain the uses of Terraform CLI ?

The Terraform Command-Line Interface (CLI) is used to manage infrastructure and interact with Terraform state, configuration files, providers, etc.

1. List some basic Terraform CLI commands?

Here are some basic CLI commands:

* terraform init - prepares your working directory for other commands
* terraform destroy - destroys the previously-created infrastructure
* terraform validate - check whether the configuration is valid
* terraform apply - creates or updates the infrastructure
* terraform plan - shows changes needed by the current configuration

1. Does Terraform support multi-provider deployments?

Yes, multi-provider deployments are supported by Terraform

1. How is duplicate resource error ignored during terraform apply?

We can try the following options:

* Delete those resources from the cloud provider(API) and recreate them using Terraform
* Delete those resources from Terraform code to stop its management with it
* Carry out a terraform import of the resource and remove the code that is trying to recreate them

12. What are some of the built-in provisioners available in Terraform?

Here is the list of built-in provisioners in Terraform:

* Salt-masterless Provisioner
* Remote-exec Provisioner
* Puppet Provisioner
* Local-exec Provisioner
* Habitat Provisioner
* File Provisioner
* Chef Provisioner

13. Which command destroys Terraform managed infrastructure?

terraform destroy [options]