1. What is Terraform and how it is different from other IaC tools?

2. How do you call a main.tf module?

3. What exactly is Sentinel?

4. You have a Terraform configuration file that defines an infrastructure deployment. However, there are multiple instances of the same resource that need to be created. How would you modify the configuration file to achieve this?

5. You want to know from which paths Terraform is loading providers referenced in your Terraform configuration (\*.tf files). You need to enable debug messages to find this out. Which of the following would achieve this?

A. Set the environment variable TF\_LOG=TRACE

B. Set verbose logging for each provider in your Terraform configuration

C. Set the environment variable TF\_VAR\_log=TRACE

D. Set the environment variable TF\_LOG\_PATH

6. Below command will destroy everything that is being created in the infrastructure. Tell us how would you save any particular resource while destroying the complete infrastructure.

terraform destroy

7. Which module is used to store the .tfstate file in S3?

8. How do you manage sensitive data in Terraform, such as API keys or passwords?

9. You are working on a Terraform project that needs to provision an S3 bucket, and a user with read and write access to the bucket. What resources would you use to accomplish this, and how would you configure them?

10. Who maintains Terraform providers?

1. Terraform is an open-source declarative infrastructure-as-code tool that aims to provision and manage infrastructure across different environments. Unlike other IaC tools, Terraform is vendor-neutral.
2. A Terraform module (usually the root configuration module) can call other modules to include their resources in the configuration. A module referred to by another module is named child module.
3. Policies are rules that Terraform Cloud enforces on runs. You use the Sentinel policy language to define Sentinel policies. After you define policies, you must add them to policy sets that Terraform Cloud can enforce on workspaces.
4. When a module has multiple configurations for the same provider, which meta-argument can you use to specify the configuration? The correct answer is identified as “provider”. There are two main reasons to use the providers argument: Using different default provider configurations for a child module.
5. Set the environment variable TF\_LOG=TRACE
6. To prevent destroy operations for specific resources, you can add the prevent\_destroy attribute to your resource definition. This lifecycle option prevents Terraform from accidentally removing critical resources. Add prevent\_destroy to your EC2 instance. Run terraform destroy to observe the behavior.
7. Terraform Backend
8. The general ways for properly handling secrets in TF are:

use specialized, external vaults, such as Terraform Vault, AWS Parameter Store or AWS Secret Manger. ...

use local-exec to setup the secrets outside of TF. ...

if the above solutions are not accessible, then you have to protect your state file (its good practice anyway).

1. How to Create an S3 Bucket Using Terraform: An Example

Step 1: Create the bucket.tf File. ...

Step 2: Create the variables.tf File. ...

Step 3: Execute the Commands in Terraform. ...

Step 1: Create the bucket.tf File. ...

Step 2: Create the variables.tf File. ...

Step 3: Execute the Commands in Terraform.

1. Terraform providers are published and maintained by a variety of sources, including HashiCorp, HashiCorp Technology Partners, and the Terraform community. The Registry uses tiers and badges to denote the source of a provider.