**Common Terraform Interview Questions**  
These questions cover a range of Terraform concepts and are designed to assess both basic and more advanced knowledge of the tool.  
  
🔷 What is Terraform, and how does it work?  
  
🔸 Answer:   
Terraform is an open-source Infrastructure as Code (IaC) tool used for provisioning and managing infrastructure. It allows users to define infrastructure in a declarative configuration language and then apply those configurations to create or update infrastructure.  
  
🔷 Explain the importance of Terraform state.  
  
🔸 Answer:   
Terraform state is a record of the infrastructure's current state. It helps Terraform understand what resources were created and what changes are pending. Storing state is crucial for collaboration and preventing conflicts in a team environment.  
  
🔷 How does Terraform handle dependencies between resources?  
  
🔸 Answer:   
Terraform automatically handles dependencies between resources by creating a dependency graph. It ensures resources are created or updated in the correct order based on their dependencies.  
  
🔷 What is the purpose of Terraform variables?  
  
🔸 Answer:   
Terraform variables are used to parameterize configurations. They allow users to input values dynamically, making configurations more flexible and reusable across environments.  
  
🔷 Explain the difference between Terraform apply and plan commands.  
  
🔸 Answer:   
The terraform plan command shows the execution plan without making any changes, while terraform apply applies the changes described in the plan. plan is for previewing changes, and apply is for executing those changes.  
  
🔷 How can sensitive data, like API keys, be managed securely in Terraform?  
  
🔸 Answer:   
Sensitive data can be managed using Terraform's sensitive input variables or by using external tools like HashiCorp Vault. Avoid hardcoding sensitive information in configuration files.  
  
🔷 What is a Terraform module, and why would you use it?  
  
🔸 Answer:   
A Terraform module is a collection of Terraform configurations grouped together. It promotes code reuse, encapsulation, and abstraction. Modules are useful for organizing and structuring Terraform code.  
  
🔷 Explain the purpose of Terraform provisioners.  
  
🔸 Answer:   
Terraform provisioners are used to execute scripts or commands on a local or remote machine as part of resource creation or updates. They are typically used for tasks like configuration management or bootstrapping.  
  
🔷 How can Terraform work with multiple environments (e.g., development, staging, production)?  
  
🔸 Answer:   
Terraform can use variables and workspaces to manage multiple environments. Variables can be used to parameterize configurations, while workspaces allow you to create separate instances of state for each environment.