Basic Questions

Q1. What is terraform?

**Terraform** is an open-source **Infrastructure as Code (IaC)** tool developed by **HashiCorp**. It allows you to **provision, manage, and version cloud infrastructure** using a declarative configuration language called **HashiCorp Configuration Language (HCL)**.

Q2. What is Infrastructure as Code (IaC)?

**Infrastructure as Code (IaC)** is the practice of managing and provisioning computing infrastructure — like servers, networks, and databases — **through machine-readable configuration files**, rather than through manual processes or interactive configuration tools (like clicking around in a cloud console).

Q3. What are Terraform providers?

**Terraform providers** basically ek **bridge hote hain** Terraform aur external systems ke beech. Jaise agar mujhe Azure pe server banana hai, to main Azure provider use karta hoon. Ye provider Terraform ko batata hai ki usko Azure ke sath kaise connect karna hai aur kaunse resources create karne hain."

Q4. What is a Terraform Configuration File?

A **Terraform configuration file** is a plain text file (usually with the .tf extension) where you **write code to define your infrastructure** using **HCL** (HashiCorp Configuration Language).

Q5. What is the purpose of the terraform. tfstate file?

The terraform. tfstate file is **very important** in Terraform. It keeps track of the **real infrastructure** that Terraform has created.

"terraform.tfstate ek file hai jisme Terraform record rakhta hai ki kaunse resources bana chuka hai, unki current settings kya hain, aur unka status kya hai. Ye file Terraform ko yaad dilati hai ki kya-kya bana hai taaki wo changes ko track kar sake."

Intermediate Questions

Q1. What are Terraform variables? How do you define and use them?

**Terraform variables** are a way to **make your code reusable, flexible, and cleaner**.

Types of Terraform Variables:

**Input Variables** – values you pass into the configuration

**Output Variables** – values you get *out* after deployment

**Environment Variables** – values from OS environment (optional)

Q2. What Are Output Values in Terraform?

**Output values** in Terraform are used to **display or share important information** after your infrastructure is created — like IP addresses, instance IDs, URLs, etc.

Q3. What is a data source in Terraform?

A **data source** in Terraform is used to **Fetch** or **read existing information** from your cloud provider **without creating** anything new.

Q4. How can you import existing infrastructure into Terraform?

Using terraform import command

Steps: Write code for resources first into terraform then run terraform import <resource ID>

Q5. What are the local’s blocks?

**locals block** ek aisi jagah hai jahan hum apni **temporary values** rakhte hain, jo hum apne Terraform code ke andar baar-baar use karna chahte hain.

Maan lo ki tumhara naam baar-baar code mein likhna hai, toh ek baar locals mein likh lo, fir jahan chahiye bas uska naam use kar lo.

locals {

Environment = "Development "

}

Q6. How do you manage sensitive values (like passwords or keys) in Terraform?

Using Azure key vaults to store our sensitive information like secret (username & PW), keys and certificates.

Q7. What is a module in Terraform, and why should you use it?

We are using modules in terraform to call the resources from the child module into the parent module.

Q8. How do you structure a Terraform project for reusability and scalability?

* Use modules
* Separate environments (Dev, QA, UAT, PROD)
* Keep variables, outputs, and main.tf separately

Q9. What are workspaces in Terraform? When would you use them?

Terraform workspace ka matlab hai — **ek hi code se alag-alag environment ka infrastructure banana**, lekin sabka **alag state file** hona.

Create Workspace: terraform workspace new dev