50 Terraform Mock Interview Questions & Answers (DevOps)

1. Q1. What is Terraform and why is it used in DevOps?

A: Terraform is an Infrastructure as Code (IaC) tool that provisions and manages infrastructure declaratively. In DevOps, it automates infrastructure provisioning.

2. Q2. What is the difference between Terraform and Ansible?

A: Terraform is mainly for provisioning infrastructure, while Ansible is primarily for configuration management.

3. Q3. What does declarative language mean in Terraform?

A: Declarative means you define the desired state, and Terraform automatically figures out how to achieve it.

4. Q4. Which language is Terraform written in?

A: Go language.

5. Q5. Which language is used to write Terraform configurations?

A: HCL (HashiCorp Configuration Language).

6. Q6. What are the main features of Terraform?

A: IaC, Immutable infrastructure, Idempotency, Multi-cloud support, Dependency resolution.

7. Q7. Difference between terraform plan and terraform apply?

A: terraform plan previews the changes, terraform apply executes the changes.

8. Q8. When do you use terraform destroy?

A: To completely remove the infrastructure.

9. Q9. What is Terraform state file?

A: A file (terraform.tfstate) that stores the current state of infrastructure.

10. Q10. Where can Terraform state files be stored?

A: Locally or remotely (S3, GCS, Azure Blob, Terraform Cloud).

11. Q11. What is a Terraform provider?

A: A plugin that allows Terraform to interact with APIs (AWS, Azure, GCP, Kubernetes).

12. Q12. What is a Terraform module?

A: A reusable collection of Terraform configurations.

13. Q13. How do you define variables in Terraform?

A: Using the variable block.

14. Q14. What are Terraform outputs used for?

A: To expose useful information from the infrastructure.

15. Q15. How does Terraform handle dependencies?

A: Automatically using resource references.

16. Q16. What is a Terraform remote backend?

A: A remote storage for state files.

17. Q17. What is the use of workspaces in Terraform?

A: To manage multiple environments (dev, stage, prod).

18. Q18. What does terraform graph command do?

A: Generates a visual dependency graph of resources.

19. Q19. What does terraform refresh do?

A: Updates state file with real infrastructure.

20. Q20. What do terraform taint and untaint do?

A: taint marks a resource for recreation, untaint removes that mark.

◆ **Basic Questions**

1. **What is Terraform and why is it used in DevOps?**
    - Terraform is an Infrastructure as Code (IaC) tool that provisions and manages infrastructure declaratively. In DevOps, it automates infrastructure provisioning.
2. **What is the difference between Terraform and Ansible?**
    - Terraform is mainly for provisioning infrastructure, while Ansible is primarily for configuration management.
3. **What does declarative language mean in Terraform?**
    - Declarative means you define the desired state, and Terraform automatically figures out how to achieve it.
4. **Which language is Terraform written in?**
    - Go language.
5. **Which language is used to write Terraform configurations?**
    - HCL (HashiCorp Configuration Language).
6. **What are the main features of Terraform?**
    - IaC, Immutable infrastructure, Idempotency, Multi-cloud support, Dependency resolution.
7. **Difference between terraform plan and terraform apply?**
    - terraform plan previews the changes, terraform apply executes the changes.
8. **When do you use terraform destroy?**

   - To completely remove the infrastructure.

9. **What is Terraform state file?**
   - A file (terraform.tfstate) that stores the current state of infrastructure.
10. **Where can Terraform state files be stored?**
    - Locally or remotely (S3, GCS, Azure Blob, Terraform Cloud).

◆ Intermediate Questions

11. **What is a Terraform provider?**
    - A plugin that allows Terraform to interact with APIs (AWS, Azure, GCP, Kubernetes).
12. **What is a Terraform module?**
    - A reusable collection of Terraform configurations.
13. **How do you define variables in Terraform?**
    - Using the variable block. Example:
    - variable "region" {
    -   default = "us-east-1"
    - }
14. **What are Terraform outputs used for?**
    - To expose useful information from the infrastructure (e.g., instance IP).
15. **How does Terraform handle dependencies?**
    - Automatically using resource references.
16. **What is a Terraform remote backend?**
    - A remote storage for state files (e.g., AWS S3).
17. **What is the use of workspaces in Terraform?**
    - To manage multiple environments (dev, stage, prod).
18. **What does terraform graph command do?**

- Generates a visual dependency graph of resources.

19. **What does terraform refresh do?**
    - Updates state file with real infrastructure.
20. **What do terraform taint and untaint do?**
    - taint marks a resource for recreation, untaint removes that mark.

◆ **Advanced Questions**

21. **What are Terraform lifecycle rules?**
    - Rules like create_before_destroy, prevent_destroy that control resource behavior.
22. **What is drift in Terraform?**
    - When actual infrastructure differs from the state file.
23. **What are provisioners in Terraform?**
    - Scripts run on resources (e.g., remote-exec, local-exec).
24. **How do you manage sensitive variables in Terraform?**
    - Use sensitive = true and secret managers (Vault, AWS Secrets Manager).
25. **Why is state file locking important?**
    - To prevent concurrent modifications.
26. **What will you do if a state file gets corrupted?**
    - Restore from backup or re-import resources using terraform import.
27. **What is the use of terraform import?**
    - To import existing infrastructure into Terraform state.
28. **When do you use null resources in Terraform?**
    - When you only need to execute a script without creating infra.
29. **What are dynamic blocks in Terraform?**
    - Used to dynamically generate nested configurations.

30. **Difference between remote-exec and local-exec provisioners?**
    - remote-exec runs commands on a remote resource, local-exec runs on the local machine.

Scenario-Based Questions

31. **What happens if two people modify the same state file?**
    - Conflicts and overwrites. Prevented by remote backend + locking.
32. **If terraform destroy is run accidentally, how to recover?**
    - Restore from backup or re-run apply with the same configs.
33. **How to safely update infrastructure in production with Terraform?**
    - Run terraform plan, review, and apply after approvals.
34. **How can blue-green deployments be implemented in Terraform?**
    - By using workspaces or separate modules and switching traffic.
35. **How do you estimate cloud costs using Terraform?**
    - Using Infracost or Terraform Cloud Cost Estimation.
36. **Best practices for secret management in Terraform?**
    - Use Vault, AWS Secrets Manager, or Parameter Store.
37. **How do you manage multi-cloud infrastructure with Terraform?**
    - Declare multiple providers.
38. **How to write conditional expressions in Terraform?**
    - Example:
    - count = var.create ? 1 : 0
39. **Difference between count and for_each?**
    - count is index-based, for_each works with maps/sets for unique keys.
40. **How do you detect drift in Terraform?**
    - By running terraform plan.

🔹 Expert / Real-Time Questions

41. **How do you make Terraform code reusable and scalable?**
    - By using modules, variables, outputs, and remote backends.
42. **How do teams collaborate with Terraform?**
    - Using remote backends, Git, and CI/CD pipelines.
43. **What is the upgrade strategy for Terraform versions?**
    - Test in staging before production rollout.
44. **How to ensure compliance and security in Terraform?**
    - Using Sentinel, OPA, or Checkov policies.
45. **How to achieve zero-downtime deployments in Terraform?**
    - Use create_before_destroy and load balancers.
46. **What are some common Terraform errors?**
    - State lock errors, provider mismatches, drift issues.
47. **How do you automate Terraform execution?**
    - Integrate with CI/CD pipelines (Jenkins, GitHub Actions).
48. **Why is provider version pinning important in Terraform?**
    - To avoid breaking changes from provider updates.
49. **Difference between Terraform Cloud and Terraform Enterprise?**
    - Cloud = SaaS; Enterprise = self-hosted with advanced governance.
50. **What are alternatives to Terraform?**
    - Pulumi, AWS CloudFormation, Azure ARM Templates, CDK.