Here are 100 scenario-based questions for Terraform, covering a range of difficulty levels and topics:

1. \*Basic Configuration\*

1. How do you create a basic VM instance in Azure using Terraform?

2. What is the purpose of a provider block in Terraform, and how do you configure it for Azure?

3. Explain how to use variables in Terraform. Provide an example of defining and using a variable.

4. How do you output values from a Terraform module? Provide an example.

5. Describe how you would configure a VNET in Azure using Terraform.

2. \*State Management\*

1. What is Terraform state, and why is it important?

2. How do you handle sensitive information, like passwords, in Terraform state files?

3. Describe a scenario where you would use terraform import.

4. Explain the purpose of terraform refresh and when you would use it.

5. How can you recover from a corrupted state file?

3. \*Modules\*

1. What is a Terraform module, and how do you create one?

2. How do you use a module from the Terraform Registry in your configuration?

3. Provide an example of how to pass parameters to a module.

4. Describe the best practices for structuring Terraform modules.

5. Explain how to use module versioning in Terraform.

4. \*Provisioners\*

1. When would you use a provisioner in Terraform?

2. Describe the different types of provisioners available in Terraform.

3. Provide an example of using a local-exec provisioner.

4. Explain how to handle provisioner failures.

5. How do you use a remote-exec provisioner to run a script on an Azure VM instance?

5. \*Terraform Commands\*

1. Explain the purpose of terraform init.

2. Describe what happens when you run terraform plan.

3. What does the terraform apply command do, and what options can you use with it?

4. How do you roll back changes in Terraform?

5. Explain the difference between terraform taint and terraform untaint.

6. \*Backends\*

1. What is a backend in Terraform, and why would you use one?

2. Describe the steps to configure a remote backend in Terraform.

3. How do you migrate Terraform state to a remote backend?

4. Explain the use of terraform init -backend-config.

5. What are the benefits of using Terraform Cloud as a backend?

7. \*Workspaces\*

1. What are Terraform workspaces, and when should you use them?

2. How do you create a new workspace in Terraform?

3. Explain how to switch between workspaces.

4. Provide an example scenario where workspaces are useful.

5. How do workspaces affect the state file?

8. \*Data Sources\*

1. What is a data source in Terraform, and how is it used?

2. Provide an example of using a data source to retrieve information about an AWS VPC.

3. Explain how data sources can be used to enhance reusability in modules.

4. How do you use a data source to get information about an existing resource?

5. What are the limitations of using data sources in Terraform?

9. \*Terraform Registry\*

1. How do you find and use modules from the Terraform Registry?

2. Describe how to publish a module to the Terraform Registry.

3. Explain the benefits of using modules from the Terraform Registry.

4. How do you version a module on the Terraform Registry?

5. Provide an example of using a third-party module from the Terraform Registry.

10. \*Advanced Features\*

1. What are dynamic blocks in Terraform, and how do you use them?

2. Explain how to use for\_each and count in resource definitions.

3. Describe the benefits and use cases for Terraform's moved block.

4. How do you use Terraform's locals to simplify complex configurations?

5. Provide an example of using conditional expressions in Terraform.

11. \*Security and Compliance\*

1. How do you manage secrets in Terraform configurations?

2. Describe how to implement role-based access control (RBAC) with Terraform.

3. What are the best practices for securing Terraform state files?

4. Explain how to use Terraform Sentinel for policy enforcement.

5. How do you audit changes made by Terraform?

12. \*Terraform Enterprise/Cloud\*

1. What are the key features of Terraform Enterprise?

2. How do you set up a workspace in Terraform Cloud?

3. Explain the use of VCS-driven workflows in Terraform Cloud.

4. Describe how to manage Sentinel policies in Terraform Cloud.

5. What are the benefits of using Terraform Cloud's cost estimation feature?

13. \*Integrations\*

1. How do you integrate Terraform with Jenkins for CI/CD?

2. Describe how to use Terraform with Ansible for configuration management.

3. How do you use Terraform with Kubernetes?

4. Explain the process of integrating Terraform with Packer for immutable infrastructure.

5. How do you use Terraform to provision resources in multiple cloud providers?

14. \*Best Practices\*

1. Describe the best practices for organizing Terraform code.

2. Explain the concept of "drift" and how to manage it in Terraform.

3. How do you handle resource dependencies in Terraform?

4. What are the advantages of using Terraform's state locking feature?

5. Provide an example of using Terraform to enforce tagging policies.

15. \*Performance Optimization\*

1. How can you optimize the performance of Terraform apply?

2. Describe how to use parallelism to speed up Terraform operations.

3. What are the trade-offs of using larger count values in Terraform?

4. How do you use Terraform's caching mechanisms?

5. Explain the impact of module nesting on Terraform performance.

These questions cover a broad spectrum of Terraform functionalities and scenarios, helping to solidify understanding and practical application of Terraform in real-world environments.