

107 Terraform interview questions for your next candidate

Questions

1. What is Terraform, in simple terms, and why do we use it?
2. Can you explain what Infrastructure as Code (IaC) means, and how Terraform fits into that?
3. What are the basic Terraform commands you would use to create infrastructure?
4. What is a Terraform provider, and can you name a few common ones?
5. What is a Terraform module, and why might you use one?
6. What is the purpose of the Terraform state file?
7. How would you apply changes to your infrastructure using Terraform?
8. Explain what happens when you run `terraform init`.
9. Explain what happens when you run `terraform plan`.
10. Explain what happens when you run `terraform apply`.
11. What is the difference between `terraform plan` and `terraform apply`?
12. If you make a mistake in your Terraform code, how can you fix it?
13. What are variables in Terraform, and why are they useful?
14. What are outputs in Terraform, and what are they used for?
15. Can you describe a situation where you might need to use Terraform?
16. What is the purpose of version control systems like Git when working with Terraform?
17. How does Terraform help with collaboration when multiple people are working on the same infrastructure?
18. What are some best practices to follow when writing Terraform code?
19. How would you define resources in Terraform to create a simple virtual machine?
20. What are the advantages of using Terraform over manually creating infrastructure?
21. What is the use of `terraform destroy` command?
22. Name the different types of provisioners in Terraform.
23. How would you handle sensitive data, like passwords, in your Terraform configuration?
24. Explain the concept of idempotence in the context of Terraform.
25. How would you update an existing resource using Terraform?
26. Explain the importance of testing your Terraform code. How would you approach testing?
27. What is Terraform, in the simplest terms?
28. Imagine you want to build a Lego house. How would Terraform help you do that?
29. What are the basic building blocks of a Terraform configuration?
30. What is the purpose of a Terraform provider?
31. Can you explain what a Terraform resource is?
32. What does 'infrastructure as code' mean to you?
33. What is Terraform's state file and why is it important?
34. How would you initialize a Terraform project?
35. What is the command to apply a Terraform configuration?
36. How do you see what Terraform *will* do before actually doing it?
37. What is the command to undo the changes made by Terraform?
38. What is a Terraform module and why would you use one?
39. How can you pass values into a Terraform module?
40. What is a variable in Terraform and why is it useful?
41. How can you define different environments in Terraform (like dev, staging, prod)?
42. What is an output in Terraform and how do you use it?
43. What does it mean when Terraform says it's 'refreshing state'?
44. Can you describe a situation where Terraform might fail?
45. How would you find documentation for a specific Terraform resource?
46. What's the difference between 'terraform apply' and 'terraform destroy'?
47. Why would you want to store your Terraform state remotely?
48. If you made a mistake in your Terraform code, how would you fix it?
49. Can you name a cloud provider that Terraform supports?
50. What is the purpose of version control, such as Git, in Terraform projects?
51. How would you handle sensitive information, like passwords, in your Terraform code?
52. What are some benefits of using Terraform over manually configuring infrastructure?
53. Explain the difference between declarative and imperative approaches to infrastructure management. Which one does Terraform follow?
54. What is the role of input variables in Terraform modules, and how can they improve reusability?
55. Describe a scenario where you might use Terraform to automate a simple task, such as creating a virtual machine.
56. How would you handle sensitive data, like passwords, in your Terraform configurations, especially when storing them in a version control system?
57. Explain how Terraform Cloud or Enterprise can improve collaboration and workflow for a team using Terraform.
58. Describe a scenario where you would use a Terraform workspace and how it benefits your infrastructure management.
59. How do you manage Terraform state in a team environment to prevent conflicts and ensure consistency?
60. Explain the difference between `terraform apply`, `terraform plan`, and `terraform destroy`.
61. How would you use Terraform to provision resources across multiple cloud providers (e.g., AWS and Azure)?
62. Describe how you would implement a blue-green deployment strategy using Terraform.
63. Explain how you can use Terraform to manage the lifecycle of a Docker container.
64. How do you use Terraform modules to create reusable and maintainable infrastructure components?
65. Describe a scenario where you would use Terraform's `count` or `for_each` meta-arguments and explain why.
66. Explain how you would configure Terraform to automatically rollback infrastructure changes in case of an error during the apply process.
67. How do you handle dependencies between different Terraform modules to ensure resources are created in the correct order?
68. Describe how you would use Terraform to manage DNS records for your infrastructure.
69. Explain how you can use Terraform to provision and configure a Kubernetes cluster.
70. How would you implement infrastructure as code (IaC) testing using tools like Kitchen or Terratest?
71. Explain how you would use Terraform to manage the infrastructure for a serverless application (e.g., AWS Lambda functions).
72. How do you monitor the health and performance of your Terraform-managed infrastructure?
73. Describe how you would use Terraform to create and manage a virtual private network (VPN).
74. Explain how you can use Terraform to automate the creation of security groups and firewall rules.
75. How would you use Terraform to manage the infrastructure for a database cluster (e.g., MySQL, PostgreSQL)?
76. Describe how you would use Terraform to create and manage a load balancer.
77. Explain how you can use Terraform to automate the process of scaling your infrastructure up or down based on demand.
78. How would you use Terraform to manage the infrastructure for a CI/CD pipeline?
79. Explain the benefits of using Terraform providers and give examples of when you would use a community provider vs. an official one.
80. How can you implement version control for your Terraform modules and configurations?
81. Describe a time when you had to troubleshoot a complex Terraform deployment. What steps did you take to diagnose and resolve the issue?
82. Explain Terraform's state locking and why it's important in a team environment.
83. How would you implement a blue/green deployment strategy using Terraform?
84. Describe a situation where you had to debug a complex Terraform configuration. What steps did you take?
85. How do you manage sensitive data, like passwords or API keys, in your Terraform configurations?
86. Explain the difference between Terraform refresh, plan, and apply commands. When would you use each?
87. How do you handle Terraform state management in a large, distributed team?
88. Describe a time when you had to roll back a Terraform deployment. What challenges did you face?
89. How would you use Terraform to manage resources across multiple cloud providers?
90. Explain the concept of Terraform workspaces and their use cases.
91. How do you ensure idempotency in your Terraform modules?
92. Describe how you would use Terraform to automate the creation of a CI/CD pipeline.
93. Explain the purpose of Terraform providers and how they interact with cloud APIs.
94. How do you handle dependencies between different Terraform modules?
95. Describe a situation where you had to optimize a Terraform configuration for performance.
96. How would you implement Terraform Cloud or Enterprise in an organization? What are the benefits?
97. Explain how to use Terraform to create and manage DNS records.
98. How do you approach testing Terraform code? What tools or techniques do you use?
99. Describe a time when you had to contribute to an existing Terraform codebase. What challenges did you encounter?
100. How would you use Terraform to create and manage a virtual network?
101. Explain the use of Terraform dynamic blocks.
102. How do you manage Terraform module versions and dependencies?
103. Describe a situation where you had to troubleshoot a Terraform provider issue.
104. How would you use Terraform to automate the creation of an auto-scaling group?
105. Explain the concept of Terraform data sources and how they are used.