Terraform Top Interview Questions

1	Explain the term Infrastructure as Code (IAC), and How terraform help in the context of infrastructure automation?
2	What are the various advantages of using Terraform over traditional infrastructure provisioning methods?
3	What do you meant by Terraform state file, and why is it important?
4	Explain the difference between declarative and imperative approaches in infrastructure provisioning with example?
5	How can you specify dependencies between resources in Terraform?
6	What is the purpose of the Terraform Init, plan, Apply command?
7	What are Terraform variables, and how can you use them in your infrastructure code?
8	How does Terraform handle secrets and sensitive data?
9	What are Terraform backends, and how do they help in state management?
10	Explain the difference between Terraform's local and remote backends.
11	What is the purpose of the "Terraform init" command?
12	What is the purpose of the "Terraform get" command?
13	What is modules in terraform? How does Terraform handle dependencies between modules?
14	What are the steps involved in importing existing infrastructure into Terraform?
15	Explain the concept of Terraform workspaces and when to use them.
16	What are provider plugins in Terraform, and how do they work?
17	How you can use Terraform to provision resources in different cloud providers simultaneously.
18	What is Terraform's "force-unlock" command used for?
19	Explain how to use the Terraform "output" block for exporting resource information
20	How can you handle resource dependencies between multiple Terraform configurations?
21	How can you define multiple providers for different regions within the same configuration file?
23	Describe the purpose of Terraform's "version" constraints in module declarations.
24	Explain the benefits of using the Terraform "plan" command for infrastructure changes.
25	Describe the process of using Terraform's "remote state data" feature for cross-configuration communication.
26	What is the purpose of the Terraform "import" command?
27	How can you use Terraform to manage infrastructure across multiple cloud providers simultaneously?
28	How can you use the "depends_on" attribute in Terraform resource blocks?

29	What is the purpose of the "Terraform refresh" command, and when would you use it?
30	Describe how you can use the "Terraform state" command to manage Terraform state files.
31	Explain the concept of remote state locking in Terraform and its importance in team collaboration.
32	How can you manage infrastructure secrets securely in Terraform, such as API keys or passwords?
33	What is the purpose of Terraform modules, and how do they promote reusability?
34	Explain the concept of Terraform providers and their role in resource provisioning.
35	Describe how you can use Terraform workspaces to manage multiple environments.
36	How can you organize your Terraform codebase for better maintainability?
37	Explain the concept of Terraform backends and the available options.
38	How does Terraform handle drift detection and reconciliation?
39	Explain the difference between Terraform's "apply" and "refresh" commands.
40	What is the purpose of Terraform's "null_resource," and when would you use it?
41	Explain the concept of Terraform's "data" blocks and their use cases.
42	Explain how to use Terraform workspaces for environment-specific variable values.
43	What are Terraform workspaces, and how do they differ from Terraform environments?
44	How can you define dynamic block attributes using Terraform's "dynamic" blocks?
45	Describe the process of using the Terraform "plan-out" flag for plan output preservation.
46	How can you use Terraform with infrastructure orchestration tools like Kubernetes or Docker Swarm?
47	Describe how you can use Terraform with infrastructure deployment tools like Ansible or Chef.
48	How can you perform blue-green deployments using Terraform?
49	How can you integrate Terraform with CI/CD pipelines?
50	How can you leverage the Terraform Graph command for visualizing resource dependencies?
51	Explain the purpose of the Terraform "taint" and "untaint" commands.