Terraform Quickstart

Quiz and 5 Hands-On Project Ideas

By Kirshi Yin
Curious Devs Corner

Quiz

Q1. What is a key benefit of Infrastructure as Code (IaC)?
A) You can avoid writing any documentation
B) You manage infrastructure manually through scripts
C) You define and automate infrastructure using code
D) It eliminates the need for version control

Q2. Which command must be run first in a new Terraform project?
A) `terraform plan`
B) `terraform fmt`
C) `terraform init`
D) `terraform apply`

Q3. What does a Terraform provider do?
A) It stores the Terraform state
B) It defines input variables
C) It connects Terraform to external APIs
D) It sets resource dependencies

Q4. What is wrong with this code snippet?

```
provider "aws" {
  region = us-east-1
}
```

A) The region name is invalid
B) The region is missing quotes
C) The provider block is not allowed here
D) Terraform does not support AWS

Q5. Which step should run before `terraform apply` in a CI/CD pipeline?
A) `terraform state show`
B) `terraform import`
C) `terraform plan`
D) `terraform destroy`

Q6. What's a main benefit of using Terraform in CI/CD?
A) It removes the need for any configuration files
B) It lets you manage infrastructure changes automatically
C) It installs providers automatically
D) It removes the need for state files

Q7. What is stored in the Terraform state file?
A) Only outputs
B) Only variables
C) The full infrastructure's current status
D) The Terraform binary

Q8. Why is using a remote backend preferred for teams?
A) It stores resources in memory
B) It removes the need for version control
C) It allows encrypted modules
D) It avoids state conflicts and supports locking

Q9. What is the purpose of a module in Terraform?
A) It replaces the state file
B) It creates a cloud database
C) It groups reusable configuration
D) It defines environment variables

Q10. How do you reuse a module in a project?
A) Use `module` blocks with `source` pointing to the module path
B) Copy the module into your main file
C) Use `provider` blocks inside the module
D) Import it using `terraform import`

Q11. What are workspaces used for in Terraform?
A) Storing provider credentials
B) Managing multiple state files for different environments
C) Creating isolated VPCs
D) Storing GitHub Actions workflows

Q12. Which command creates a new workspace?
A) `terraform new`
B) `terraform workspace add`

C) `terraform workspace new dev`
D) `terraform switch dev`

Q13. What does `terraform validate` do?
A) Applies your infrastructure
B) Runs unit tests
C) Checks for syntax and configuration errors
D) Formats your Terraform files

Q14. What's the benefit of using Terratest?
A) It deploys production infrastructure instantly
B) It applies all modules in one shot
C) It allows writing automated tests for Terraform modules
D) It replaces the state file

Q15. What does `terraform import` do?
A) Automatically creates Terraform code for you
B) Migrates state to a remote backend
C) Adds an existing resource into Terraform state
D) Validates imported code

Q16. What does setting `TF_LOG=DEBUG` help with?
A) Encrypts all state output
B) Enables provider tracing
C) Shows detailed internal logs during execution
D) Disables backend logging

Answer Sheet

Q# Answer

1 C

2 C

3 C

4 B

5 C

6 B

7 C

8 D

9 C

10 A

11 B

12 C

13 C

14 C

15 C

16 C

5 Hands-On Projects Using Free Providers

1. Create a Static Website with GitHub Pages

- Provider: `integrations/github`
- What you'll do: create a GitHub repo, configure Pages settings, and push a static site template.

2. Set Up GitHub Branch Protection Rules

- Provider: `integrations/github`
- What you'll do: manage protected branches, required reviews, and CODEOWNERS via Terraform.

3. Provision a Free PostgreSQL Database on Render

- Provider: `render-oss/render`
- What you'll do: provision a managed database, retrieve credentials, and optionally connect to a Flask app.

4. Create a CI/CD Workflow with GitHub Actions

- Provider: `integrations/github`
- What you'll do: define a `.github/workflows` file using `github_repository_file` and manage it with Terraform.

5. Manage Teams and Repositories for an Open Source Project

- Provider: `integrations/github`
- What you'll do: create a full GitHub org structure — teams, repos, permissions — defined in code.