Or it will be maintainable, as long as you follow the Golden Rule of Terraform.

The Golden Rule of Terraform

Here's a quick way to check the health of your Terraform code: go into your *live* repository, pick several folders at random, and run `terraform plan` in each one. If the output is always "no changes," that's great, because it means that your infrastructure code matches what's actually deployed. If the output sometimes shows a small diff, and you hear the occasional excuse from your team members ("Oh, right, I tweaked that one thing by hand and forgot to update the code"), your code doesn't match reality, and you might soon be in trouble. If `terraform plan` fails completely with weird errors, or every `plan` shows a gigantic diff, your Terraform code has no relation at all to reality and is likely useless.

The gold standard, or what you're really aiming for, is what I call *The Golden Rule of Terraform*:

> *The main branch of the live repository should be a 1:1 representation of what's actually deployed in production.*

Let's break this sentence down, starting at the end and working our way back:

"...what's actually deployed"

> The only way to ensure that the Terraform code in the *live* repository is an up-to-date representation of what's actually deployed is to *never make out-of-band changes*. After you begin using Terraform, do not make changes via a web UI, or manual API calls, or any other mechanism. As you saw in Chapter 5, out-of-band changes not only lead to complicated bugs, but they also void many of the benefits you get from using IaC in the first place.

"...a 1:1 representation..."