- What are the best practices for using Terraform as a team?

The only tools you need are a computer (Terraform runs on most operating systems), an internet connection, and the desire to learn.

Why I Wrote This Book

Terraform is a powerful tool. It works with all popular cloud providers. It uses a clean, simple language and has strong support for reuse, testing, and versioning. It's open source and has a friendly, active community. But there is one area where it's lacking: maturity.

Terraform has become wildly popular, but it's still a relatively new technology, and despite its popularity, it's still difficult to find books, blog posts, or experts to help you become proficient with the tool. The official Terraform documentation does a good job of introducing the basic syntax and features, but it includes little information on idiomatic patterns, best practices, testing, reusability, or team workflows. It's like trying to become fluent in French by studying only the vocabulary but not any of the grammar or idioms.

The reason I wrote this book is to help developers become fluent in Terraform. I've been using Terraform for six out of the seven years it has existed, mostly at my company, Gruntwork, where Terraform is one of the core tools we've used to create a library of more than 300,000 lines of reusable, battle-tested infrastructure code that's used in production by hundreds of companies. Writing and maintaining this much infrastructure code over this many years and using it with so many different companies and use cases has taught us a lot of hard lessons. My goal is to share these lessons with you so that you can cut this lengthy process down and become fluent in a matter of days.

Of course, you can't become fluent just by reading. To become fluent in French, you need to spend time conversing with native French speakers, watching French TV shows, and listening to French music. To become fluent in Terraform, you need to write real Terraform code, use it to manage