

real software, and deploy that software on real servers. Therefore, be ready to read, write, and execute a lot of code.

## What You Will Find in This Book

Here's an outline of what the book covers:

### *Chapter 1, “Why Terraform”*

How DevOps is transforming the way we run software; an overview of infrastructure-as-code tools, including configuration management, server templating, orchestration, and provisioning tools; the benefits of infrastructure as code; a comparison of Terraform, Chef, Puppet, Ansible, Pulumi, OpenStack Heat, and CloudFormation; how to combine tools such as Terraform, Packer, Docker, Ansible, and Kubernetes.

### *Chapter 2, “Getting Started with Terraform”*

Installing Terraform; an overview of Terraform syntax; an overview of the Terraform CLI tool; how to deploy a single server; how to deploy a web server; how to deploy a cluster of web servers; how to deploy a load balancer; how to clean up resources you've created.

### *Chapter 3, “How to Manage Terraform State”*

What Terraform state is; how to store state so that multiple team members can access it; how to lock state files to prevent race conditions; how to isolate state files to limit the damage from errors; how to use Terraform workspaces; a best-practices file and folder layout for Terraform projects; how to use read-only state.

### *Chapter 4, “How to Create Reusable Infrastructure with Terraform Modules”*

What modules are; how to create a basic module; how to make a module configurable with inputs and outputs; local values; versioned

modules; module gotchas; using modules to define reusable, configurable pieces of infrastructure.

### *Chapter 5, “Terraform Tips and Tricks: Loops, If-Statements, Deployment, and Gotchas”*

Loops with the `count` parameter, `for_each` and `for` expressions, and the `for` string directive; conditionals with the `count` parameter, `for_each` and `for` expressions, and the `if` string directive; built-in functions; zero-downtime deployment; common Terraform gotchas and pitfalls, including `count` and `for_each` limitations, zero-downtime deployment gotchas, how valid plans can fail, and how to refactor Terraform code safely.

### *Chapter 6, “Managing Secrets with Terraform”*

An introduction to secrets management; an overview of the different types of secrets, different ways to store secrets, and different ways to access secrets; a comparison of common secret management tools such as HashiCorp Vault, AWS Secrets Manager, and Azure Key Vault; how to manage secrets when working with providers, including authentication via environment variables, IAM roles, and OIDC; how to manage secrets when working with resources and data sources, including how to use environment variables, encrypted files, and centralized secret stores; how to securely handle state files and plan files.

### *Chapter 7, “Working with Multiple Providers”*

A closer look at how Terraform providers work, including how to install them, how to control the version, and how to use them in your code; how to use multiple copies of the same provider, including how to deploy to multiple AWS regions, how to deploy to multiple AWS accounts, and how to build reusable modules that can use multiple providers; how to use multiple different providers together, including an

example of using Terraform to run a Kubernetes cluster (EKS) in AWS and deploy Dockerized apps into the cluster.

### *Chapter 8, “Production-Grade Terraform Code”*

Why DevOps projects always take longer than you expect; the production-grade infrastructure checklist; how to build Terraform modules for production; small modules; composable modules; testable modules; reusable modules; Terraform Registry; variable validation; versioning Terraform, Terraform providers, Terraform modules, and Terragrunt; Terraform escape hatches.

### *Chapter 9, “How to Test Terraform Code”*

Manual tests for Terraform code; sandbox environments and cleanup; automated tests for Terraform code; Terratest; unit tests; integration tests; end-to-end tests; dependency injection; running tests in parallel; test stages; retries; the test pyramid; static analysis; plan testing; server testing.

### *Chapter 10, “How to Use Terraform as a Team”*

How to adopt Terraform as a team; how to convince your boss; a workflow for deploying application code; a workflow for deploying infrastructure code; version control; the golden rule of Terraform; code reviews; coding guidelines; Terraform style; CI/CD for Terraform; the deployment process.

Feel free to read the book from beginning to end or jump around to the chapters that interest you the most. Note that the examples in each chapter reference and build upon the examples from the previous chapters, so if you skip around, use the open source code examples (as described in “[Open Source Code Examples](#)”) to get your bearings. At the end of the book, in the [Appendix A](#), you’ll find a list of recommended reading where you can learn more about Terraform, operations, IaC, and DevOps.