**Preface**

A long time ago, in a data center far, far away, an ancient group of powerful beings known as sysadmins used to deploy infrastructure manually. Every server, every database, every load balancer, and every bit of network configuration was created and managed by hand. It was a dark and fearful age: fear of downtime, fear of accidental misconfiguration, fear of slow and fragile deployments, and fear of what would happen if the sysadmins fell to the dark side (i.e., took a vacation). The good news is that thanks to the DevOps movement, there is now a better way to do things: Terraform.

[Terraform](https://www.terraform.io/) is an open source tool created by HashiCorp that allows you to define your infrastructure as code using a simple, declarative programming language, and to deploy and manage that infrastructure across a variety of public cloud providers (e.g., Amazon Web Services, Azure, Google Cloud, DigitalOcean) and private cloud and virtualization platforms (e.g., OpenStack, VMWare) using a few commands. For example, instead of manually clicking around a web page or running dozens of commands, here is all the code it takes to configure a server on Amazon Web Services:

**provider** "aws" {

region = "us-east-1"

}

**resource** "aws\_instance" "example" {

ami = "ami-40d28157"

instance\_type = "t2.micro"

}

And to deploy it, you just run one command:

> terraform apply

Thanks to its simplicity and power, Terraform is rapidly emerging as a key player in the DevOps world. It allows you to replace the tedious, fragile, and manual parts of sysadmin work with a solid, automated foundation upon which you can build all your other DevOps practices (e.g., automated testing, continuous integration, continuous delivery) and tooling (e.g., Docker, Chef, Puppet).

This book is the fastest way to get up and running with Terraform.

You’ll go from deploying the most basic “Hello, World” Terraform example (in fact, you just saw it!) all the way up to running a full tech stack (server cluster, load balancer, database) capable of supporting a large amount of traffic and a large team of developers—all in the span of just a few chapters. This is a hands-on tutorial that not only teaches you DevOps and infrastructure as code principles, but also walks you through dozens of code examples that you can try at home, so make sure you have your computer handy.

By the time you’re done, you’ll be ready to use Terraform in the real world.

**Who Should Read This Book**

This book is for anyone responsible for the code after it has been written. That includes Sysadmins, Operations Engineers, Release Engineers, Site Reliability Engineers, DevOps Engineers, Infrastructure Developers, Full Stack Developers, Engineering Managers, and CTOs. No matter what your title is, if you’re the one managing infrastructure, deploying code, configuring servers, scaling clusters, backing up data, monitoring apps, and responding to alerts at 3 a.m., then this book is for you.

Collectively, all of these tasks are usually referred to as “operations.” In the past, it was common to find developers who knew how to write code, but did not understand operations; likewise, it was common to find sysadmins who understood operations, but did not know how to write code. You could get away with that divide in the past, but in the modern world, as cloud computing and the DevOps movement become ubiquitous, just about every developer will need to learn operational skills and every sysadmin will need to learn coding skills.

This book does not assume you’re already an expert coder or expert sysadmin—a basic familiarity with programming, the command line, and server-based software (e.g., websites) should suffice. Everything else you need you’ll be able to pick up as you go, so that by the end of the book, you will have a solid grasp of one of the most critical aspects of modern development and operations: managing infrastructure as code.

In fact, you’ll learn not only how to manage infrastructure as code using Terraform, but also how this fits into the overall DevOps world. Here are some of the questions you’ll be able to answer by the end of the book:

* Why use infrastructure as code at all?
* What are the differences between configuration management, provisioning, and server templating?
* When should you use Terraform, Chef, Ansible, Puppet, Salt, CloudFormation, Docker, or Packer?
* How does Terraform work and how do you use it to manage your infrastructure?
* How do you make Terraform a part of your automated deployment process?
* How do you make Terraform a part of your automated testing process?
* 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 with 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: age.

At the time of writing, Terraform is barely two years old. As a result, it’s hard to find books, blog posts, or experts to help you master the tool. If you try to learn Terraform from the official documentation, you’ll find that it does a good job of introducing the basic syntax and features, but it includes almost no 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 and 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 more than half of its life, much of it in a professional context at my company [Gruntwork](http://www.gruntwork.io/), and I’ve spent many of those months figuring out what works and what doesn’t primarily through trial and error. My goal is to share what I’ve learned so you can avoid that lengthy process and become fluent in a matter of hours.

Of course, you can’t become fluent just by reading. To become fluent in French, you’ll have to spend time talking with native French speakers, watching French TV shows, and listening to French music. To become fluent in Terraform, you’ll have to write real Terraform code, use it to manage 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,](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch01.html#why_terraform) *[Why Terraform](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch01.html#why_terraform)*

How DevOps is transforming the way we run software; an overview of infrastructure as code tools, including configuration management, provisioning, and server templating; the benefits of infrastructure as code; a comparison of Terraform, Chef, Puppet, Ansible, SaltStack, OpenStack Heat, and CloudFormation.

[Chapter 2,](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch02.html#an_intro_to_terraform_syntax) *[Getting Started with Terraform](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch02.html#an_intro_to_terraform_syntax)*

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,](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch03.html#how_to_manage_terraform_state) *[How to Manage Terraform State](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch03.html#how_to_manage_terraform_state)*

What is Terraform state; how to store state so 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; a best-practices file and folder layout for Terraform projects; how to use read-only state.

[Chapter 4,](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch04.html#how_to_create_reusable_infrastructure) *[How to Create Reusable Infrastructure with Terraform Modules](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch04.html#how_to_create_reusable_infrastructure)*

What are modules; how to create a basic module; how to make a module configurable; versioned modules; module tips and tricks; using modules to define reusable, configurable pieces of infrastructure.

[Chapter 5,](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch05.html#terraform_tips_tricks) *[Terraform Tips and Tricks: Loops, If-Statements, Deployment, and Gotchas](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch05.html#terraform_tips_tricks)*

Advanced Terraform syntax; loops; if-statements; if-else statements; interpolation functions; zero-downtime deployment; common Terraform gotchas and pitfalls.

[Chapter 6,](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch06.html#how_to_use_terraform_team) *[How to Use Terraform as a Team](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch06.html#how_to_use_terraform_team)*

Version control; the golden rule of Terraform; coding guidelines; Terraform style; automated testing for Terraform; documentation; a workflow for teams; automation with Terraform.

Feel free to read the book from start to finish or jump around to the chapters that interest you the most. At the end of the book, in [Appendix A](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/app01.html#recommended_reading), you’ll find a list of recommended reading where you can learn more about Terraform, operations, infrastructure as code, and DevOps.

**What You Won’t Find in This Book**

This book is not meant to be an exhaustive reference manual for Terraform. I do not cover all the cloud providers, or all of the resources supported by each cloud provider, or every available Terraform command. For these nitty-gritty details, I refer you instead to the [Terraform documentation](https://www.terraform.io/docs/index.html).

The documentation contains many useful answers, but if you’re new to Terraform, infrastructure as code, or operations, you won’t even know what questions to ask. Therefore, this book is focused on what the documentation does *not* cover: namely, how to go beyond introductory examples and use Terraform in a real-world setting. My goal is to get you up and running quickly by discussing why you may want to use Terraform in the first place, how to fit it into your workflow, and what practices and patterns tend to work best.

To demonstrate these patterns, I’ve included a number of code examples. I’ve tried to make it as easy as possible for you to try these examples at home by minimizing dependencies on any third parties. This is why almost all the examples use just a single cloud provider, Amazon Web Services (AWS), so you only have to sign up for a single third-party service (also, AWS offers a generous free tier, so running the example code shouldn’t cost you anything). This is why the book and the example code do not cover or require HashiCorp’s paid services, Terraform Pro and Terraform Enterprise. And this is why I’ve released all the code examples as open source.

**Open Source Code Examples**

All of the code samples in the book can be found at the following URL:

* *<https://github.com/brikis98/terraform-up-and-running-code>*

You may want to check out this repo before you start reading so you can follow along with all the examples on your own computer:

git clone https://github.com/brikis98/terraform-up-and-running-code.git

The code examples in that repo are broken down chapter by chapter. It’s worth noting that most of the examples show you what the code looks like at the *end* of a chapter. If you want to maximize your learning, you’re better off writing the code yourself, from scratch.

You’ll start coding in [Chapter 2](https://www.safaribooksonline.com/library/view/terraform-up-and/9781491977071/ch02.html#an_intro_to_terraform_syntax), where you’ll learn how to use Terraform to deploy a basic cluster of web servers from scratch. After that, follow the instructions in each subsequent chapter on how to evolve and improve this web server cluster example. Make the changes as instructed, try to write all the code yourself, and only use the sample code in the GitHub repo as a way to check your work or get yourself unstuck.

**A NOTE ABOUT VERSIONS**

All of the examples in this book were tested against Terraform 0.8.x, which was the most recent major release at the time of writing. Since Terraform is a relatively new tool and has not hit version 1.0.0 yet, it is possible that future releases will contain backward incompatible changes, and it is likely that some of the best practices will change and evolve over time.

I’ll try to release updates as often as I can, but the Terraform project moves fast, so you’ll have to do some work to keep up with it on your own. For the latest news, blog posts, and talks on Terraform and DevOps, be sure to check out this [book’s website](http://www.terraformupandrunning.com/) and subscribe to the [newsletter](http://www.terraformupandrunning.com/#newsletter)!

**Using the Code Examples**

This book is here to help you get your job done and you are welcome to use the sample code in your programs and documentation. You do not need to contact O’Reilly for permission unless you’re reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O’Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product’s documentation does require permission.

Attribution is appreciated, but not required. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Terraform: Up and Running* by Yevgeniy Brikman (O’Reilly). Copyright 2017 Yevgeniy Brikman, 978-1-491-97708-8.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact O’Reilly Media at *[permissions@oreilly.com](mailto:permissions@oreilly.com)*.

**Conventions Used in This Book**

The following typographical conventions are used in this book:

*Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

**Constant width bold**

Shows commands or other text that should be typed literally by the user.

*Constant width italic*

Shows text that should be replaced with user-supplied values or by values determined by context.

**TIP**

This element signifies a tip or suggestion.

**NOTE**

This element signifies a general note.

**WARNING**

This element indicates a warning or caution.

**O’Reilly Safari**

*[Safari](http://oreilly.com/safari)* (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O’Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, and Course Technology, among others.

For more information, please visit *<http://oreilly.com/safari>*.

**How to Contact O’Reilly Media**

Please address comments and questions concerning this book to the publisher:

* O’Reilly Media, Inc.
* 1005 Gravenstein Highway North
* Sebastopol, CA 95472
* 800-998-9938 (in the United States or Canada)
* 707-829-0515 (international or local)
* 707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *<http://bit.ly/terraform-up-and-running>*.

To comment or ask technical questions about this book, send email to *[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)*.

For more information about our books, courses, conferences, and news, see our website at *[http://www.oreilly.com](http://www.oreilly.com/)*.

Find us on Facebook: *<http://facebook.com/oreilly>*

Follow us on Twitter: *<http://twitter.com/oreillymedia>*

Watch us on YouTube: *<http://www.youtube.com/oreillymedia>*

**Acknowledgments**

Josh Padnick

This book would not have been possible without you. You were the one who introduced me to Terraform in the first place, taught me all the basics, and helped me figure out all the advanced parts. Thank you for supporting me while I took our collective learnings and turned them into a book. Thank you for being an awesome cofounder and making it possible to run a startup while still living a fun life. And thank you most of all for being a good friend and a good person.

O’Reilly Media

Thank you for publishing another one of my books. Reading and writing have profoundly transformed my life and I’m proud to have your help in sharing some of my writing with others. A special thanks to Brian Anderson for helping me get this book out in record time.

Gruntwork customers

Thank you for taking a chance on a small, unknown company, and volunteering to be guinea pigs for our Terraform experiments. Gruntwork’s mission is to make it an order of magnitude easier to understand, develop, and deploy software. We haven’t always succeeded at that mission (I’ve captured many of our mistakes in this book!), so I’m grateful for your patience and willingness to be part of our audacious attempt to improve the world of software.

HashiCorp

Thank you for building an amazing collection of DevOps tools, including Terraform, Packer, Consul, and Vault. You’ve improved the world of DevOps and with it, the lives of millions of software developers.

Kief Morris, Seth Vargo, Mattias Gees

Thank you for reading an early version of this book and providing lots of detailed, constructive feedback. Your suggestions have made this book significantly better.

Mom, Dad, Larisa, Molly

I accidentally wrote another book. That probably means I didn’t spend as much time with you as I wanted. Thank you for putting up with me anyway. I love you.