

# *about this book*

---

The goal of this book is to help turn chaos engineering into a mature, mainstream, science-based practice, accessible to anyone. I strongly believe that it might offer some of the best return on investment you can get, and I want everyone to be able to benefit from that.

One of the challenges of writing a book like this is that chaos engineering doesn't focus on any single technology or programming language. In fact, it can be used on all kinds of stacks, which is one of its advantages. You can see that reflected in this book—each chapter is focused on a popular situation a software engineer might find themselves in, dealing with different languages, layers of the stack, and levels of control over the source code. The book uses Linux as the primary operating system, but the principles it teaches are universal.

## ***Who should read this book***

This book is for anyone who wants to make their systems more reliable. Are you an SRE? A full-stack developer? Frontend developer? Do you work with JVM, containers, or Kubernetes? If you said yes to any of these, you will find chapters of this book written for you. The book assumes a minimal familiarity with running day-to-day commands on Linux (Ubuntu). This is not an introduction to all of these things, and I assume a basic understanding of them so that we can dive deep (notable exceptions are Docker and Kubernetes, which are relatively new technologies, and we do cover how they work first).

## How this book is organized: a roadmap

The book ships 13 chapters, split across three parts.

After chapter 1 introduces chaos engineering and the reasons for implementing it, part 1 lays the groundwork for further understanding what chaos engineering is about:

- Chapter 2 shows a real-world example of how a seemingly simple application might break in unexpected ways.
- Chapter 3 covers observability and all the tools that you’re going to need to look under the hood of your system.
- Chapter 4 takes a popular application (WordPress) and shows you how to design, execute, and analyze a chaos experiment on the networking layer.

Part 2 covers various technologies and stacks where chaos engineering shines:

- Chapter 5 takes you from a vague idea of what Docker is, to understanding how it works under the hood and testing its limitations using chaos engineering.
- Chapter 6 demystifies system calls—what they are, how to see applications make them, and how to block them to see how resistant to failure these applications are.
- Chapter 7 shows how to inject failure on the fly into the JVM, so that you can test how a complex application handles the types of failure you’re interested in.
- Chapter 8 discusses baking failure directly into your application.
- Chapter 9 covers chaos engineering . . . in the browser (using JavaScript).

Part 3 is dedicated to Kubernetes:

- Chapter 10 introduces Kubernetes, where it came from, and what it can do for you.
- Chapter 11 covers some higher-level tools that let you implement sophisticated chaos engineering experiments quickly.
- Chapter 12 takes you deep down the rabbit hole of how Kubernetes works under the hood. To understand its weak points, you need to know how it works. This chapter covers all the components that together make Kubernetes tick, along with ideas on how to identify resiliency problems using chaos engineering.

Finally, the last chapter talks about chaos engineering beyond the machines:

- Chapter 13 shows that the same principles also apply to the other complex distributed systems that you deal with on a daily basis—human teams. It covers the chaos engineering mindset, gives you ideas for games you can use to make your teams more reliable, and discusses how to get buy-in from stakeholders.

## About the code

The book contains various snippets of code along with the expected output to teach you how to use different tools. The best way to run them is to use the Ubuntu VM that ships with this book. You can download it, as well as all the source code, from <https://github.com/seeker89/chaos-engineering-book>.

### ***liveBook discussion forum***

Purchase of Chaos Engineering includes free access to a private web forum run by Manning Publications, where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To access the forum, go to <http://mng.bz/5jEO>. You can also learn more about Manning's forums and the rules of conduct at <https://livebook.manning.com/#!/discussion>.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It is not a commitment to any specific amount of participation on the part of the author, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions lest his interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.