

preface

People often ask how I ended up doing chaos engineering. I tend to tell them that I needed a sleeping aid. And chaos engineering is vegan-friendly and surprisingly effective for that purpose. Let me explain.

Back in 2016, through a lucky coincidence, I started working on a cutting-edge project based on Kubernetes. Nobody gets fired for choosing Kubernetes in 2020, but back then it was rather risky. Kubernetes v1.2 came as a bunch of moving parts, and bug fixes were rolling out quicker than we could install them.

To make it work, my team needed to build real operational experience, and do it fast. We needed to know how things worked and broke, how to fix them, and how to get alerted when that happened. And the best way to do that, we reasoned, was to break them preemptively.

This practice, which I later learned to call *chaos engineering* for the extra cool factor, turned out to be very effective at reducing the number of outages. And that, in turn, was better for my sleep quality than the expensive, bamboo-coated, memory foam pillow I have. Fast-forward a few years, and chaos engineering is one of my primary interests. And I'm not alone—it is quickly becoming an invaluable tool to engineers around the world.

Today chaos engineering suffers from a few serious problems. In particular, the urban myths (that it's about randomly breaking things in production), a lack of quality content that teaches people how to do it well, and the initially counterintuitive mindset that needs to be adopted (failure will happen, so we need to be ready).

I wrote this book to fix these problems. I want to move chaos engineering from the funky zone to a legitimate, science-based methodology that's applicable to any system,

software or otherwise. I want to show that you don't need to have massive scale to benefit from it, and that it can give you a lot of value for a little investment.

This book is designed for all curious software engineers and developers who want to build more reliable systems, however tiny or humongous they might be. And it gives them the right tools, from the Linux kernel all the way up to the application or browser level.

I've put a lot of work into making this book what it is now, and I'm hoping that you get value—and a few laughs—out of it. And finally, let's stay in touch. If you'd like to hear more from me, subscribe to my newsletter at <https://chaosengineering.news>. And if you like (or hate) the book, reach out and tell me all about it!