Chapter 1. Why Terraform

Software isn't done when the code is working on your computer. It's not done when the tests pass. And it's not done when someone gives you a "ship it" on a code review. Software isn't done until you *deliver* it to the user.

Software delivery consists of all of the work you need to do to make the code available to a customer, such as running that code on production servers, making the code resilient to outages and traffic spikes, and protecting the code from attackers. Before you dive into the details of Terraform, it's worth taking a step back to see where Terraform fits into the bigger picture of software delivery.

In this chapter, you'll dive into the following topics:

- What is DevOps?
- What is infrastructure as code?
- What are the benefits of infrastructure as code?
- How does Terraform work?
- How does Terraform compare to other infrastructure-as-code tools?

What Is DevOps?

In the not-so-distant past, if you wanted to build a software company, you also needed to manage a lot of hardware. You would set up cabinets and racks, load them up with servers, hook up wiring, install cooling, build redundant power systems, and so on. It made sense to have one team, typically called Developers ("Devs"), dedicated to writing the software, and a separate team, typically called Operations ("Ops"), dedicated to managing this hardware.