



Gorgo Tasting OpenTofu and Terraform

September 08, 2026

Chapter 0

How to use this booklet

This is a short booklet to give you a taste of OpenTofu. By the end you will have a web server that serves sayings, backed by a database, running first on your machine, then on Kubernetes, and finally on the internet with a real name and a real certificate. All of it is created, changed, and torn down with one command.

Each chapter is meant to help you understand a topic, but you will still want to reference the provider documentation for the specifics of every argument. Hopefully, after this taste, you will have the context you need to dig into deeper reference material.

OpenTofu and Terraform

Terraform was created by HashiCorp in 2014 and became the standard way to describe infrastructure as code. In 2023 HashiCorp changed its license to one that is no longer open source, and the community forked the last open source release as OpenTofu under the Linux Foundation. The two projects share the same language, the same providers, the same state file format, and the same commands. The differences are at the edges, and Appendix A lists the ones that matter.

This booklet treats the two as equivalent. Commands are written as `tofu`; if you use Terraform, type `terraform` instead. Every configuration in this booklet works with both, and where one has a feature the other lacks, the booklet does not use it.



Wut: The top-level configuration block is called `terraform` even in OpenTofu. OpenTofu kept the name so that existing configurations keep working. When you see `terraform { ... }` in a file, read it as “settings for OpenTofu”.

Callouts

Tips call out details that you need to pay special attention to. **Traps** warn you of common mistakes. **Wut** calls out a detail that is counter-intuitive, so make sure you pay attention.

What you need

- **OpenTofu** (or Terraform). On macOS, `brew install opentofu`. On Ubuntu, `snap install --classic opentofu`, or follow the installer instructions at opentofu.org. On Windows, `winget install OpenTofu.Tofu`. Check with `tofu version`.
- **Docker** running on your machine (Chapters 2 through 5): Docker Engine on Linux, Docker Desktop on macOS and Windows.

- **Go** 1.21 or newer (Chapters 2 and 3, for running the server outside a container): the installers at go.dev/dl cover all three systems, or `brew install go` on macOS and `winget install GoLang.Go` on Windows. The examples declare Go 1.26 in `go.mod`, and any Go from 1.21 on fetches that toolchain for you the first time it is needed.
- **kind** and **kubectrl** (Chapter 4). On macOS, `brew install kind kubectrl`. On Windows, `winget install Kubernetes.kind` and then `winget install Kubernetes.kubectrl`. On Linux, the release binaries from the two projects' sites.
- For Chapter 5, an **Oracle Cloud** free tier account, a domain whose Domain Name System (DNS) records are hosted by **Cloudflare** (free is fine), and an email address for **Let's Encrypt**.

Windows

The sessions in this booklet use a POSIX (Portable Operating System Interface) shell, which is what macOS and Linux give you. On Windows you have two good options.

The simplest is **WSL** (Windows Subsystem for Linux): install Ubuntu from the Microsoft Store, install Docker Desktop with its WSL integration turned on, and then follow the Linux instructions inside the Ubuntu terminal. Everything in the booklet works there exactly as printed.

The other is to stay in **PowerShell** with the native Windows tools, which works too, with a few translations:

In the booklet	In PowerShell
<code>export NAME=value</code>	<code>\$env:NAME = "value"</code>
<code>NAME=value tofu apply</code>	<code>\$env:NAME = "value"; tofu apply</code>
<code>\$(command)</code>	the same
<code>curl</code>	<code>curl.exe</code> (plain <code>curl</code> is an alias for a different command)
<code>command & (run in the background)</code>	run it in a second terminal
<code>~/ssh/id_ed25519.pub, ~/.kube/config</code>	the same; OpenTofu expands <code>~</code> to your home directory

OpenTofu itself, its language, and the providers behave the same on all three systems, and so do the configurations in this booklet.

Reading the examples

Configuration is shown in terraform code blocks:

```
resource "terraform_data" "hello" {
  input = "hello world"
}
```

Terminal sessions are shown in an outlined box. Bold lines that start with `$` are what you type — don't type the `$`; everything else is what the tool printed back:

```
$ tofu plan
Plan: 1 to add, 0 to change, 0 to destroy.
```

Output is trimmed to the interesting lines, and a line with only `...` stands for output that was left out. Your output will have more in it, and the ids and timings will differ.

The complete, tested configurations for every chapter are in the `examples` directory next to this booklet, under the directory names the chapters use: `hello-world`, `webserver`, `database`, `kubernetes`, `oracle`, the `shared`

modules, and motd, the sayings module that Chapters 3 to 5 pull from GitHub. Each one is self-contained, so you can cd into it and run `tofu init` and `tofu apply` without copying anything else.

Chapter layout

Each chapter follows a consistent structure:

- **The need, why it matters, why it is hard, and the strategy** at the top, so you know the problem the chapter solves before it starts solving it.
- The **main content** with explanations, configuration, commands, and callouts.
- **Key Points** summarizing the concepts that were introduced.
- **New Syntax** collecting the language elements the chapter introduced in one table.
- **Try It** suggestions of things to change, break, and extend.
- **Exercises** with a mix of questions to test your understanding.

New syntax is introduced when a chapter needs it, with an attempt to explain why it exists and how to think about it, not just what it is.

Appendices

Appendix A collects best practices, recommendations, and common errors for OpenTofu and Terraform in one place. Skim it once now so that you know it exists, and read it properly after Chapter 3, when the advice will make sense. The chapters point to it when they run into one of its items.

Appendix B lists what OpenTofu provides without any provider: the built-in resource and data source, the block types, the meta-arguments, the named values, the operators, and the commonly used built-in functions with an example of each. Keep it open while you write configuration.

Exercises

Do not skip the exercises at the end of the chapters. You can get the answer key, but do not look at it before you work out the answer yourself. If you peek first, the concepts will not sink in. And most of all, run the examples. Infrastructure as code is learned by applying, breaking, and destroying things, and there is no other way that comes close to it.

