	Types of secrets	Secret storage	Secret interface

[a] Vault supports multiple *secret engines*, most of which are designed for infrastructure secrets, but a few support customer secrets as well.

Since this is a book about Terraform, from here on out, I'll mostly be focusing on secret management tools designed for infrastructure secrets that are accessed through an API or the CLI (although I'll mention personal secret management tools from time to time too, as those often contain the secrets you need to authenticate to the infrastructure secret tools).

Secret Management Tools with Terraform

Let's now turn to how to use these secret management tools with Terraform, going through each of the three places where your Terraform code is likely to brush up against secrets:

- Providers
- Resources and data sources
- State files and plan files

Providers

Typically, your first exposure to secrets when working with Terraform is when you have to authenticate to a provider. For example, if you want to run `terraform apply` on code that uses the AWS Provider, you'll need to first authenticate to AWS, and that typically means using your access keys, which are secrets. How should you store those secrets? And how should you make them available to Terraform?

There are many ways to answer these questions. One way you should *not* do it, even though it occasionally comes up in the Terraform documentation, is by putting secrets directly into the code, in plain text: