The parameters of many resources are immutable, so if you change them, Terraform will delete the old resource and create a new one to replace it. The documentation for each resource often specifies what happens if you change a parameter, so get used to checking the documentation. And, once again, make sure to always use the `plan` command and consider whether you should use a `create_before_destroy` strategy.

Conclusion

Although Terraform is a declarative language, it includes a large number of tools, such as variables and modules, which you saw in Chapter 4, and `count`, `for_each`, `for`, `create_before_destroy`, and built-in functions, which you saw in this chapter, that give the language a surprising amount of flexibility and expressive power. There are many permutations of the if-statement tricks shown in this chapter, so spend some time browsing the functions documentation, and let your inner hacker go wild. OK, maybe not too wild, as someone still needs to maintain your code, but just wild enough that you can create clean, beautiful APIs for your modules.

Let's now move on to Chapter 6, where I'll go over how create modules that are not only clean and beautiful but also handle secrets and sensitive data in a safe and secure manner.

---

1 Credit for this technique goes to Paul Hinze.