

Live repo and modules repo

As discussed in [Chapter 4](#), you will typically want at least two separate version control repositories for your Terraform code: one repo for modules and one repo for live infrastructure. The repository for modules is where you create your reusable, versioned modules, such as all the modules you built in the previous chapters of this book (`cluster/asg-rolling-deploy`, `data-stores/mysql`, `networking/alb`, and `services/hello-world-app`). The repository for live infrastructure defines the live infrastructure you've deployed in each environment (dev, stage, prod, etc.).

One pattern that works well is to have one infrastructure team in your company that specializes in creating reusable, robust, production-grade modules. This team can create remarkable leverage for your company by building a library of modules that implement the ideas from [Chapter 8](#); that is, each module has a composable API, is thoroughly documented (including executable documentation in the *examples* folder), has a comprehensive suite of automated tests, is versioned, and implements all of your company's requirements from the production-grade infrastructure checklist (i.e., security, compliance, scalability, high availability, monitoring, and so on).

If you build such a library (or you buy one off the shelf³), all the other teams at your company will be able to consume these modules, a bit like a service catalog, to deploy and manage their own infrastructure, without (a) each team having to spend months assembling that infrastructure from scratch or (b) the Ops team becoming a bottleneck because it must deploy and manage the infrastructure for every team. Instead, the Ops team can spend most of its time writing infrastructure code, and all of the other teams will be able to work independently, using these modules to get themselves up and running. And because every team is using the same canonical modules under the hood, as the company grows and requirements change, the Ops team can push out new versions of the modules to all teams, ensuring everything stays consistent and maintainable.