

can deploy a server with Ansible) and that provisioning tools can typically do some degree of configuration (e.g., you can run configuration scripts on each server you provision with Terraform), you typically want to pick the tool that's the best fit for your use case.

In particular, if you use server templating tools, the vast majority of your configuration management needs are already taken care of. Once you have an image created from a `Dockerfile` or Packer template, all that's left to do is provision the infrastructure for running those images. And when it comes to provisioning, a provisioning tool is going to be your best choice. In [Chapter 7](#), you'll see an example of how to use Terraform and Docker together, which is a particularly popular combination these days.

That said, if you're not using server templating tools, a good alternative is to use a configuration management and provisioning tool together. For example, a popular combination is to use Terraform to provision your servers and Ansible to configure each one.

Mutable Infrastructure Versus Immutable Infrastructure

Configuration management tools such as Chef, Puppet, and Ansible typically default to a mutable infrastructure paradigm.

For example, if you instruct Chef to install a new version of OpenSSL, it will run the software update on your existing servers, and the changes will happen in place. Over time, as you apply more and more updates, each server builds up a unique history of changes. As a result, each server becomes slightly different than all the others, leading to subtle configuration bugs that are difficult to diagnose and reproduce (this is the same configuration drift problem that happens when you manage servers manually, although it's much less problematic when using a configuration management tool). Even with automated tests, these bugs are difficult to catch; a configuration management change might work just fine on a test server, but that same change might behave differently on a production server because the production server has accumulated months of changes that aren't reflected in the test environment.