Chapter 4. How to Create Reusable Infrastructure with Terraform Modules

At the end of Chapter 3, you deployed the architecture shown in Figure 4-1.

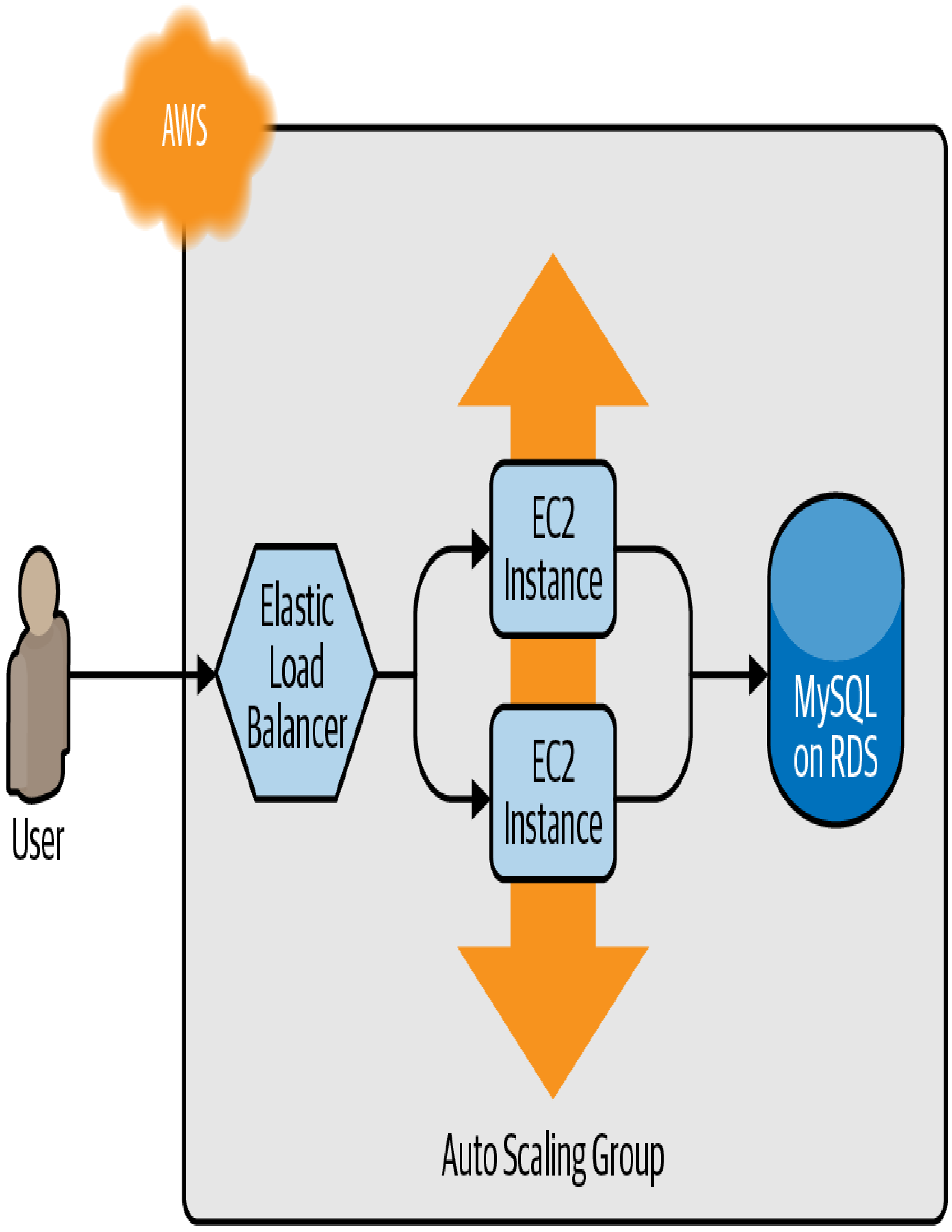


Figure 4-1. The architecture you deployed in previous chapters included a load balancer, web server cluster, and database.

This works great as a first environment, but you typically need at least two environments: one for your team’s internal testing (“staging”) and one that real users can access (“production”), as shown in Figure 4-2. Ideally, the two environments are nearly identical, though you might run slightly fewer/smaller servers in staging to save money.

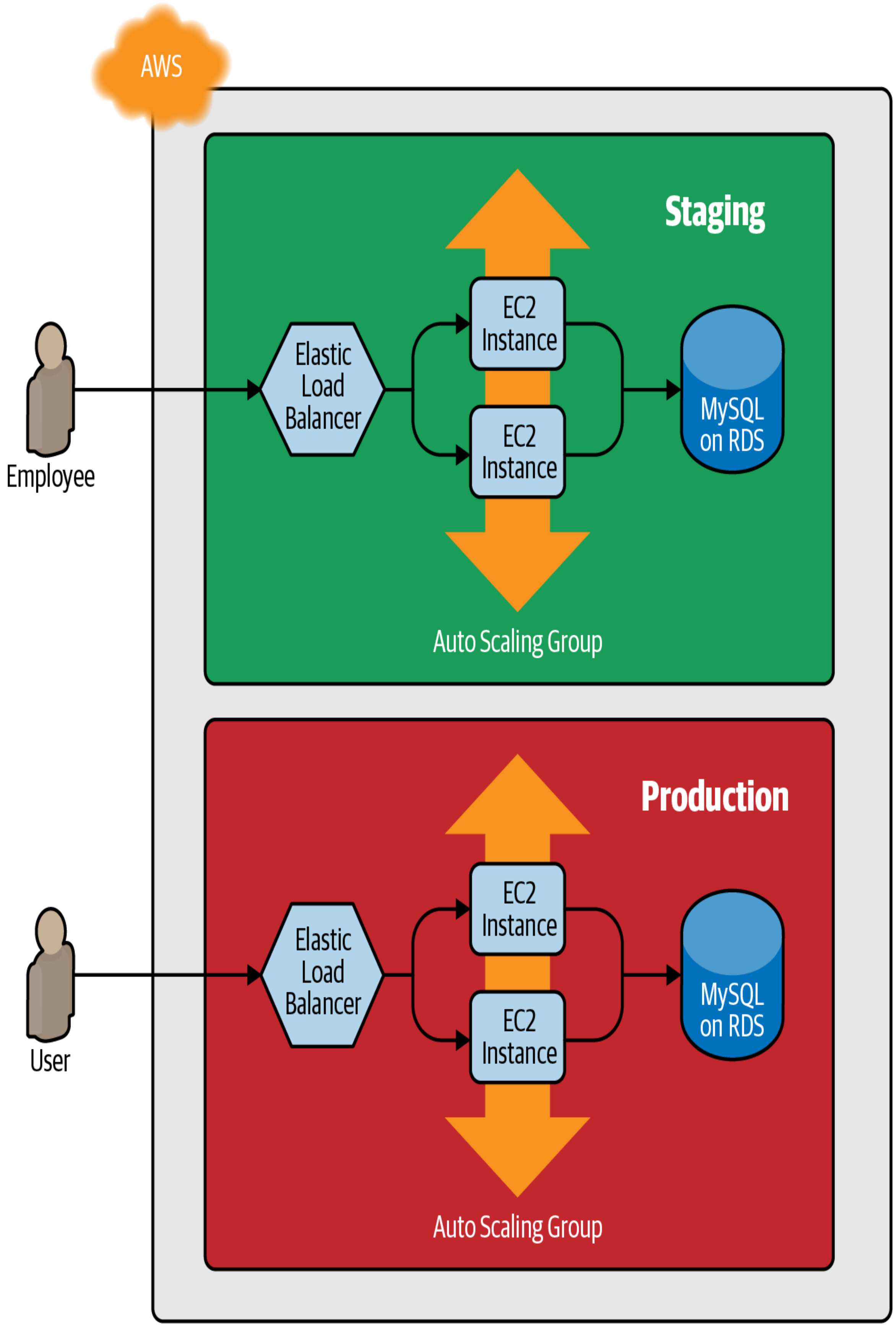

AWS
Staging
Employee
Elastic Load Balancer
EC2 Instance
EC2 Instance
MySQL on RDS
Auto Scaling Group
Production
User
Elastic Load Balancer
EC2 Instance
EC2 Instance
MySQL on RDS
Auto Scaling Group

Figure 4-2. The architecture you'll deploy in this chapter will have two environments, each with its own load balancer, web server cluster, and database.

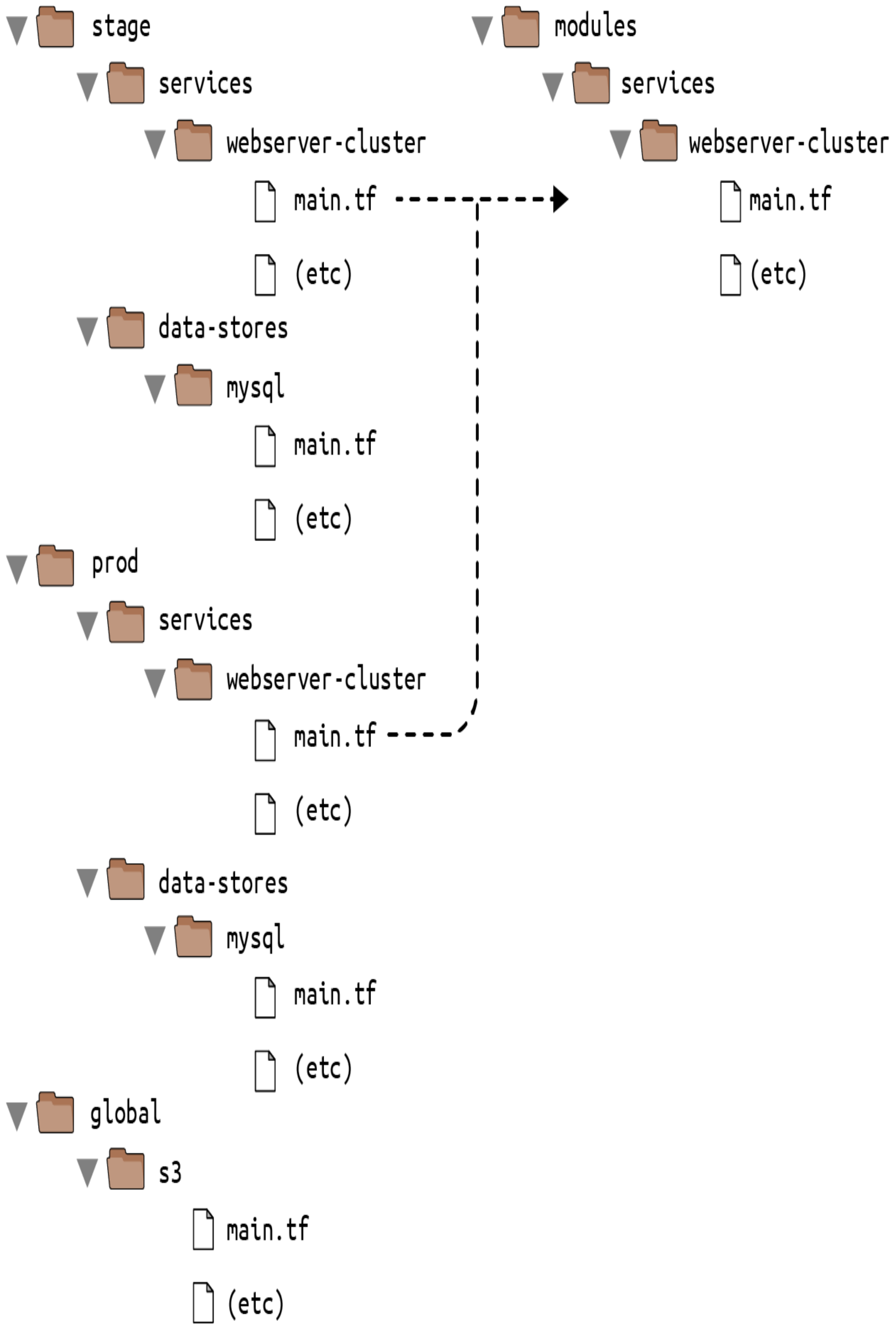
How do you add this production environment without having to copy and paste all of the code from staging? For example, how do you avoid having to copy and paste all the code in *stage/services/webserver-cluster* into *prod/services/webserver-cluster* and all the code in *stage/data-stores/mysql* into *prod/data-stores/mysql*?

In a general-purpose programming language such as Ruby, if you had the same code copied and pasted in several places, you could put that code inside of a function and reuse that function everywhere:

```
# Define the function in one place
def example_function()
  puts "Hello, World"
end

# Use the function in multiple other places
example_function()
```

With Terraform, you can put your code inside of a *Terraform module* and reuse that module in multiple places throughout your code. Instead of having the same code copied and pasted in the staging and production environments, you'll be able to have both environments reuse code from the same module, as shown in Figure 4-3.


stage
services
webserver-cluster
main.tf
(etc)
data-stores
mysql
main.tf
(etc)
prod
services
webserver-cluster
main.tf
(etc)
data-stores
mysql
main.tf
(etc)
global
s3
main.tf
(etc)
modules
services
webserver-cluster
main.tf
(etc)