```
Finished in 0.633175 seconds.
----------------------------------------
8 tests, 24 assertions, 0 failures, 0 errors
100% passed
----------------------------------------
```

The key thing to notice is that both manual and automated tests for application code can run completely locally on your own computer. You'll see later in this chapter that this is not true for the same part of the workflow for infrastructure changes.

Make Code Changes

Now that you can run the application code, you can begin making changes. This is an iterative process in which you make a change, rerun your manual or automated tests to see whether the change worked, make another change, rerun the tests, and so on.

For example, you can change the output of *web-server.rb* to "Hello, World v2," restart the server, and see the result:

```
$ curl http://localhost:8000
Hello, World v2
```

You might also update and rerun the automated tests. The idea in this part of the workflow is to optimize the feedback loop so that the time between making a change and seeing whether it worked is minimized.

As you work, you should regularly be committing your code, with clear commit messages explaining the changes you've made:

```
$ git commit -m "Updated Hello, World text"
```

Submit Changes for Review

Eventually, the code and tests will work the way you want them to, so it's time to submit your changes for a code review. You can do this with a separate code review tool (e.g., Phabricator or Review Board) or, if you're