If you deploy this cluster using `terraform apply`, wait for the Instances to register in the ALB, and open the ALB URL in a web browser, you'll see something similar to Figure 3-12.

Congrats, your web server cluster can now programmatically access the database address and port via Terraform. If you were using a real web framework (e.g., Ruby on Rails), you could set the address and port as environment variables or write them to a config file so that they could be used by your database library (e.g., ActiveRecord) to communicate with the database.

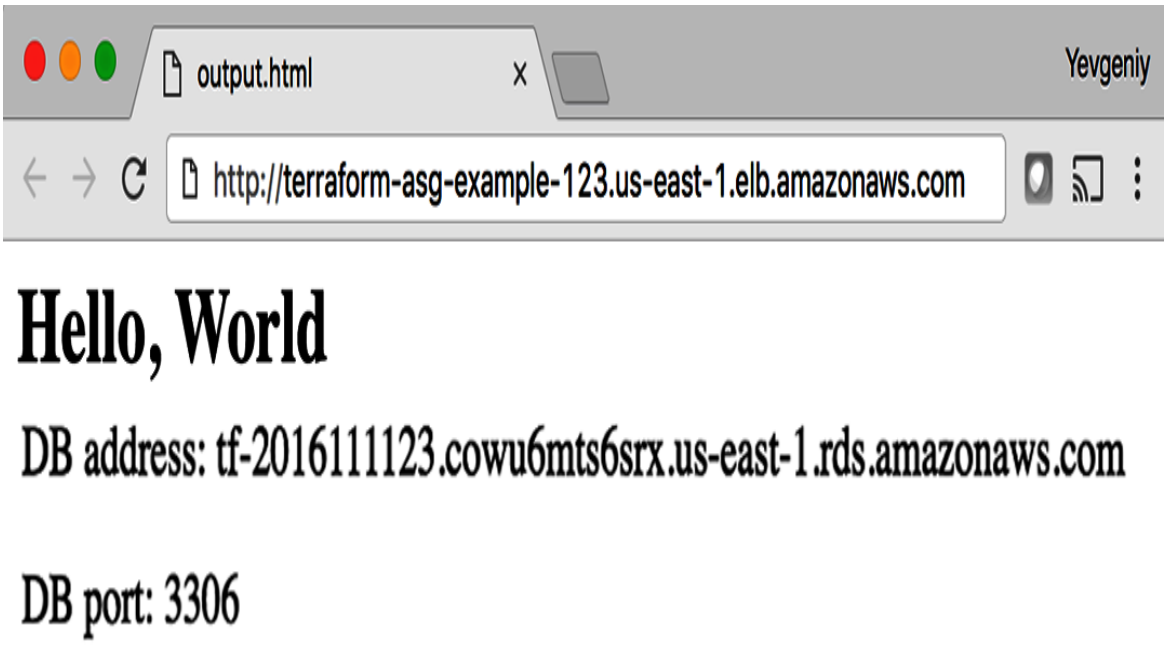

Figure 3-12. The web server cluster can programmatically access the database address and port.

Conclusion

The reason you need to put so much thought into isolation, locking, and state is that infrastructure as code (IaC) has different trade-offs than normal coding. When you're writing code for a typical app, most bugs are relatively minor and break only a small part of a single app. When you're writing code that controls your infrastructure, bugs tend to be more severe, given that they can break all of your apps—and all of your data stores, and

your entire network topology, and just about everything else. Therefore, I recommend including more "safety mechanisms" when working on IaC than with typical code.[8]

A common concern of using the recommended file layout is that it leads to code duplication. If you want to run the web server cluster in both staging and production, how do you avoid having to copy and paste a lot of code between *stage/services/webserver-cluster* and *prod/services/webserver-cluster*? The answer is that you need to use Terraform modules, which are the main topic of Chapter 4.

---

1 Learn more about S3's guarantees on the AWS website.

2 See pricing information for S3 on the AWS website.

3 Pricing information for DynamoDB is available on the AWS website.

4 Here's a colorful example of what happens when you don't isolate Terraform state.

5 The workspaces documentation makes this same exact point, but it's buried among several paragraphs of text, and as workspaces used to be called "environments," I find many users are still confused about when and when not to use workspaces.

6 You can find documentation for the `sprintf` syntax on the Go website.

7 The full list of built-in functions is available on the Terraform website.

8 For more information on software safety mechanisms, see Agility Requires Safety.