

you have to solve how to do secure authentication between multiple machines.

- Integrating with automated tests is harder, as much of the code you're testing now depends on a running, external system that either needs to be mocked out or have test data stored in it.

State Files and Plan Files

There are two more places where you'll come across secrets when using Terraform:

- State files
- Plan files

State files

Hopefully, this chapter has convinced you to not store your secrets in plain text and provided you with some better alternatives. However, something that catches many Terraform users off guard is that, no matter which technique you use, *any secrets you pass into your Terraform resources and data sources will end up in plain text in your Terraform state file!*

For example, no matter where you read the database credentials from—environment variables, encrypted files, a centralized secret store—if you pass those credentials to a resource such as `aws_db_instance`:

```
resource "aws_db_instance" "example" {
  identifier_prefix = "terraform-up-and-running"
  engine            = "mysql"
  allocated_storage = 10
  instance_class    = "db.t2.micro"
  skip_final_snapshot = true
  db_name            = var.db_name

  # Pass the secrets to the resource
  username = local.db_creds.username
  password = local.db_creds.password
}
```