```
$ terraform apply

null_resource.example (local-exec): Hello, World from Darwin
x86_64 i386
```

External data source

Provisioners will typically be your go-to for executing scripts from Terraform, but they aren’t always the correct fit. Sometimes, what you’re really looking to do is execute a script to fetch some data and make that data available within the Terraform code itself. To do this, you can use the `external` data source, which allows an external command that implements a specific protocol to act as a data source.

The protocol is as follows:

- You can pass data from Terraform to the external program using the `query` argument of the `external` data source. The external program can read in these arguments as JSON from stdin.

- The external program can pass data back to Terraform by writing JSON to stdout. The rest of your Terraform code can then pull data out of this JSON by using the `result` output attribute of the external data source.

Here’s an example:

```
data "external" "echo" {
  program = ["bash", "-c", "cat /dev/stdin"]

  query = {
    foo = "bar"
  }
}

output "echo" {
  value = data.external.echo.result
}

output "echo_foo" {
```