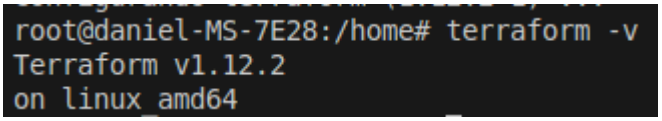
Tarea 1: Instalación de Terraform y primer script con LocalStack

1. Instalar Terraform

Utilizamos los siguientes comandos para instalar terraform en nuestro equipo:

- sudo apt-get update && sudo apt-get install -y gnupg software-properties-common curl
- curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
- echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
- sudo apt update && sudo apt install terraform

Y comprobamos que se ha instalado correctamente:



2. Instalar LocalStack

Para ello, vamos a utilizar un contenedor en docker como ya hemos hecho en prácticas anteriores:

```
root@daniel-MS-7E28:/home# docker pull localstack/localstack
Using default tag: latest
latest: Pulling from localstack/localstack
b895f377d09e: Pull complete
799440a7bae7: Pull complete
9596beeb5a6d: Pull complete
15658014cd85: Pull complete
669bfe6e51ab: Pull complete
6c45f7d01403: Pull complete
3b59a5777a73: Pull complete
c35b87a6d186: Pull complete
4f4fb700ef54: Pull complete
a24a09946db2: Pull complete
6eb26be724a8: Pull complete
6a66685ea2d4: Pull complete
3224c67f0d95: Pull complete
f7a23c721398: Pull complete
64b06d6eda51: Pull complete
f25a5bc986d7: Pull complete
1d819479d66f: Pull complete
adb5874b037c: Pull complete
85459490738d: Pull complete
e25d6e3dd84d: Pull complete
15cecc8547ad: Pull complete
0349c0b9907d: Pull complete
427ccda37fe5: Pull complete
9e362301ec40: Pull complete
Digest: sha256:a87d483c5ae99e532516bd94b40dd4e6d65562306e1d17ec1ab0dd651bc9d3eb
Status: Downloaded newer image for localstack/localstack:latest
docker.io/localstack/localstack:latest
root@daniel-MS-7E28:/home# docker run --rm -it -p 4566:4566 -p 4571:4571 localstack/localstack

LocalStack version: 4.5.1.dev62
LocalStack build date: 2025-06-27
LocalStack build git hash: 5b0180cbf

Ready.
```

3. Crear archivo `main.tf`

Creamos el archivo main.tf con el siguiente contenido en un directorio de prueba:

main.tf

terraform-localstack > main.tf

```hcl
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "= 4.67.0"
    }
  }
}

provider "aws" {
  access_key                  = "mock"
  secret_key                  = "mock"
  region                      = "us-east-1"
  s3_use_path_style           = true
  skip_credentials_validation = true
  skip_metadata_api_check     = true
  skip_requesting_account_id  = true
  endpoints {
    s3 = "http://localhost:4566"
  }
}

resource "aws_s3_bucket" "test_bucket" {
  bucket = "mi-bucket-de-prueba"
  force_destroy = true
}
```

4. Ejecutar `terraform init` y `terraform apply`

```
root@daniel-MS-7E28:/home/terraform-localstack# terraform init -upgrade
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "4.67.0"...
- Using previously-installed hashicorp/aws v4.67.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@daniel-MS-7E28:/home/terraform-localstack# terraform apply -auto-approve
aws_s3_bucket.test_bucket: Refreshing state... [id=mi-bucket-de-prueba]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
root@daniel-MS-7E28:/home/terraform-localstack#
```

5. Verificar con AWS CLI

Como ya tenemos instalado aws cli de ejercicios anteriores, basta con ejecutar el siguiente comando y comprobar que devuelve “mi-bucket-de-prueba”

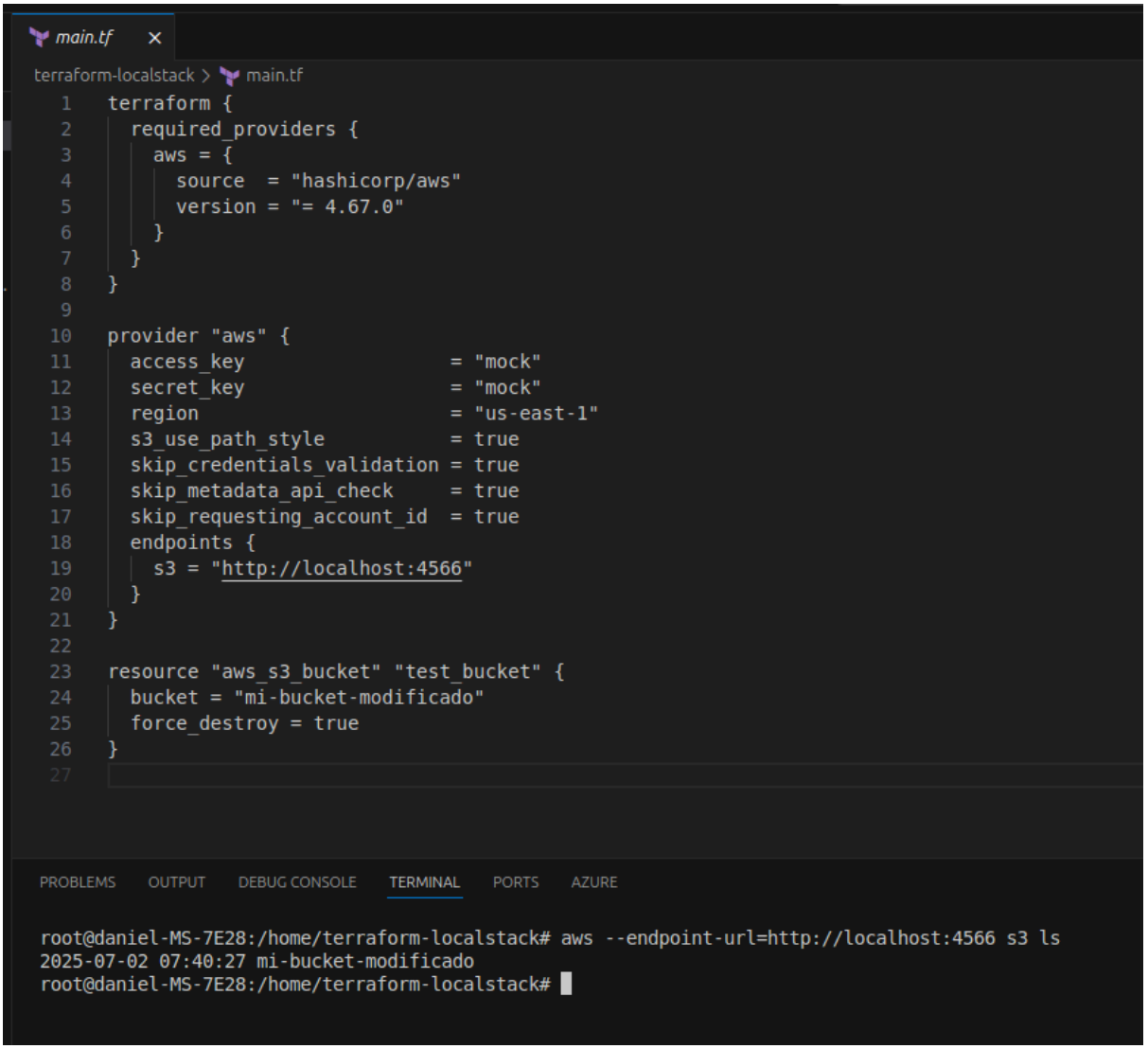
```
root@daniel-MS-7E28:/home/terraform-localstack# aws --endpoint-url=http://localhost:4566 s3 ls
2025-06-27 08:48:48 mi-bucket-de-prueba
root@daniel-MS-7E28:/home/terraform-localstack#
```

Tarea 2: Crear un recurso con Terraform usando LocalStack

En nuestro caso, ya hemos creado un recurso en la Tarea 1, resolviendo tanto la Tarea 1 como la 2 al mismo tiempo.

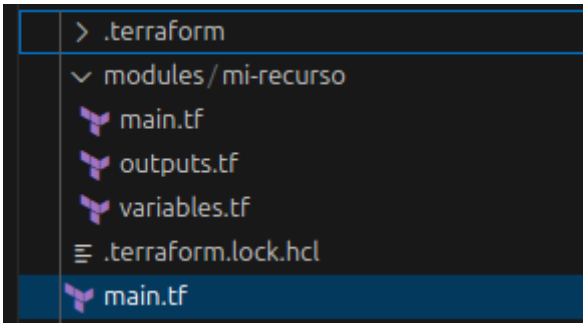
Tarea 3: Modificar un recurso usando Terraform y LocalStack

Ahora vamos a modificar el nombre del bucket creado anteriormente en el archivo main.tf
Luego comprobamos que el bucket se ha modificado correctamente.

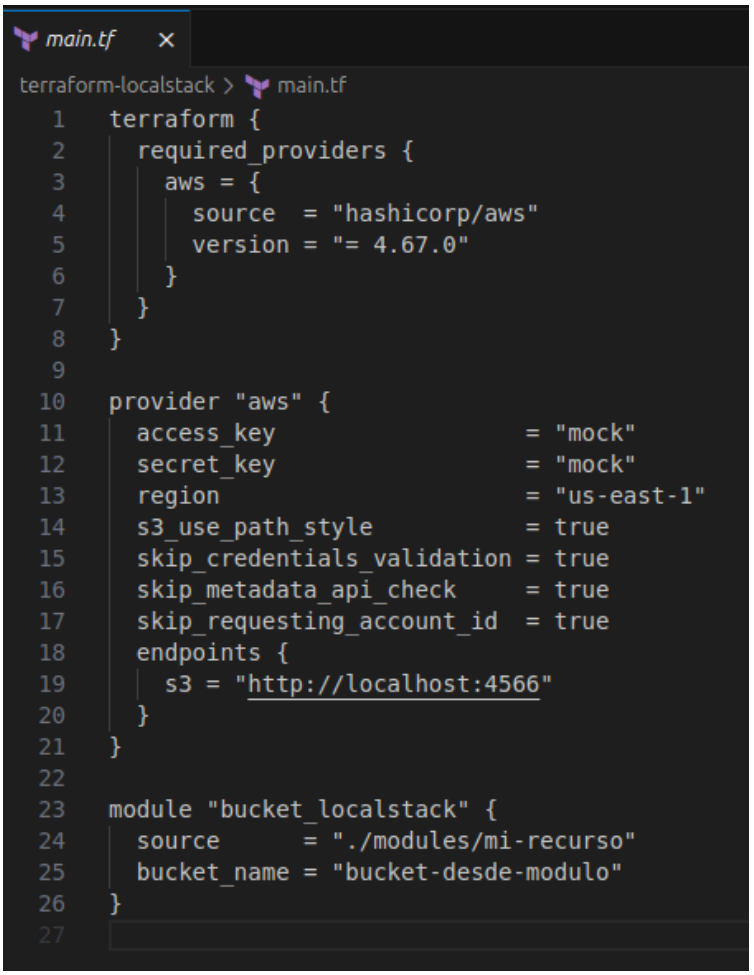


Tarea 4: Introducción a los módulos de Terraform

Estructura del proyecto:

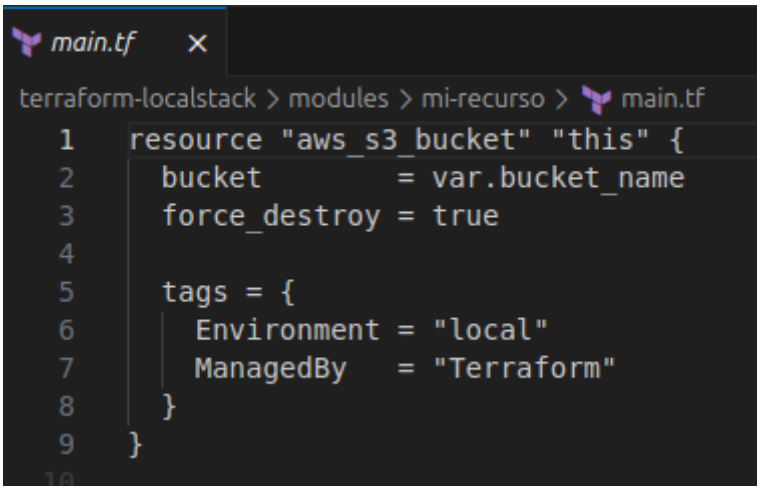


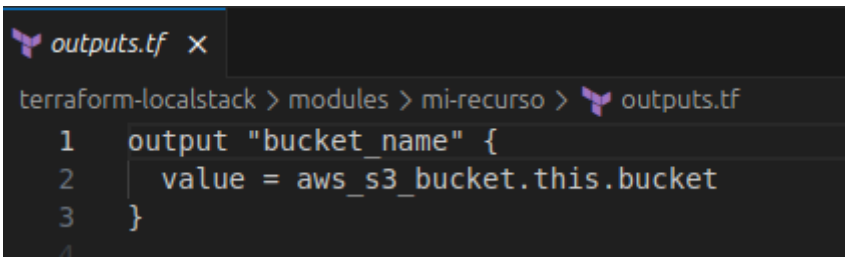
En el archivo principal, se configura el proveedor AWS apuntando a LocalStack y se invoca el módulo:

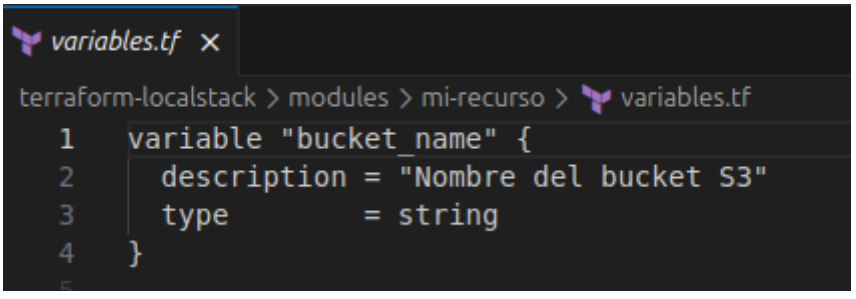


Detalles del módulo

- **Ruta:** `modules/mi-recurso/`
- **Funcionalidad:** Define un recurso `aws_s3_bucket` reutilizable.
- **Parámetros:** Recibe el nombre del bucket por variable.







Se verificó la creación del bucket con el comando: