Diplomatura en DevOps

Edición 2403

Informe Práctico Integrador N-2 (PIN-2)

"*GH Actions - apache – AWS*"

Tutor:

- Guazzardo, Marcelo

Grupo 6:

- Hector Barrios hdbarrios@gmail.com
- Juan Pablo Heyda juanpabloh.123@gmail.com
- Renzo Carletti renzocarletti@hotmail.com / pipito1498@gmail.com
- Johanna Dominguez johisd9@hotmail.com
- Lucas Bufano lucas.bufano2@gmail.com

Repositorio de GitHub Público:

- https://github.com/hdbarrios/devops-g6-pin2

Objetivos:

- Instalar terraform
- Configurar usuario programático en AWS IAM
- Desarrollar código terraform para crear una instancia EC2 aprovisionada con apache
- Utilizar github actions para desplegar un apache
    - Plus - destruir toda la infra desde github actions.

Pre- Requisitos:

Instalar terraform

https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

```
$ sudo apt-get update && sudo apt-get install -y gnupg
software-properties-common

$ wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null

$ gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint

$ 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-get install terraform -y

$ terraform -version
  Terraform v1.10.2
  on linux_amd64
```

Adicional puedes instalar `tfenv` para tener varias versiones de Terraform:

https://github.com/tfutils/tfenv

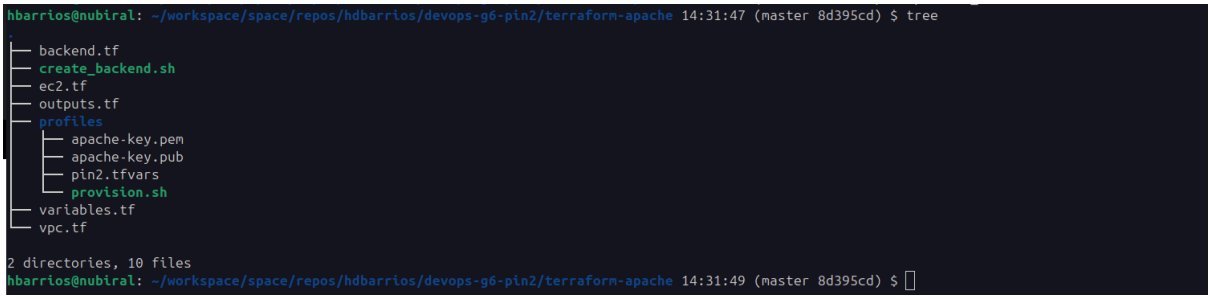
```
$ export PATH="$HOME/.tfenv/bin:$PATH"   #INCLUIR EN ~/.bash_profile
$ tfenv
tfenv 3.0.0
Usage: tfenv <command> [<options>]

Commands:
   install       Install a specific version of Terraform
   use           Switch a version to use
   uninstall     Uninstall a specific version of Terraform
   list          List all installed versions
```

```
   list-remote   List all installable versions
   version-name  Print current version
   init          Update environment to use tfenv correctly.
   pin           Write the current active version to ./.terraform-version
```

```
$ tfenv use 1.10.2
No installed versions of terraform matched '1.10.2:^1.10.2$'. Trying to
install a matching version since TFENV_AUTO_INSTALL=true
Installing Terraform v1.10.2
Downloading release tarball from
https://releases.hashicorp.com/terraform/1.10.2/terraform_1.10.2_linux_amd6
4.zip
######################################################################
######################################################################
################################## 100.0%
Downloading SHA hash file from
https://releases.hashicorp.com/terraform/1.10.2/terraform_1.10.2_SHA256SUMS
Not instructed to use Local PGP (/home/hbarrios/.tfenv/use-{gpgv,gnupg}) &
No keybase install found, skipping OpenPGP signature verification
Archive:  /tmp/tfenv_download.Yfp59d/terraform_1.10.2_linux_amd64.zip
  inflating: /home/hbarrios/.tfenv/versions/1.10.2/LICENSE.txt
  inflating: /home/hbarrios/.tfenv/versions/1.10.2/terraform
Installation of terraform v1.10.2 successful. To make this your default
version, run 'tfenv use 1.10.2'
Switching default version to v1.10.2
Default version (when not overridden by .terraform-version or
TFENV_TERRAFORM_VERSION) is now: 1.10.2
```

Crear el backend:

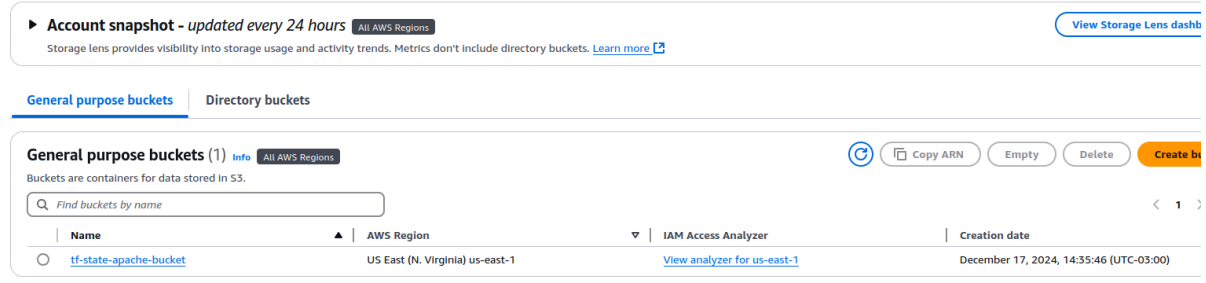
Usando **`create_backen.sh`**, si no tienes el archivo ~/.aws/config creado, ejecuta **`aws configure –profile terraform-admin`** (se aconseja si tienes más de una cuenta por administrar usar profiles)

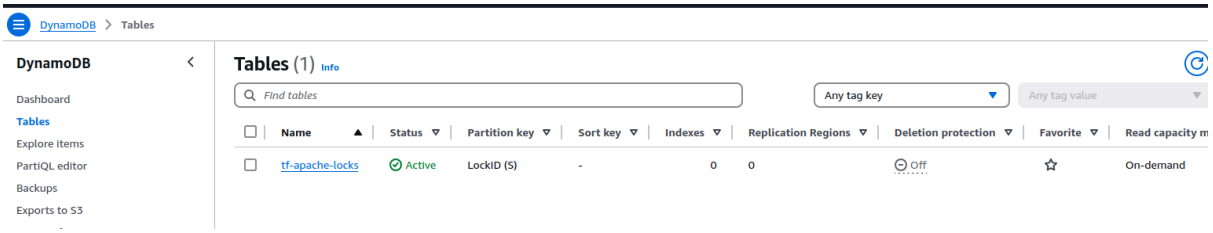


```
hbarrios@nubiral: ~/workspace/space/repos/hdbarrios/devops-g6-pin2/terraform-apache 14:35:38 (master 8d395cd) $ ./create_backend.sh
/usr/local/bin/aws
{
    "Location": "/tf-state-apache-bucket"
}
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "LockID",
                "AttributeType": "S"
            }
        ],
        "TableName": "tf-apache-locks",
        "KeySchema": [
            {
                "AttributeName": "LockID",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2024-12-17T14:35:48.078000-03:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 0,
            "WriteCapacityUnits": 0
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:536697232168:table/tf-apache-locks",
        "TableId": "157da151-dd71-4327-879f-b78396685163",
        "BillingModeSummary": {
            "BillingMode": "PAY_PER_REQUEST"
        }
    }
}
```

```
{
    "Table": {
        "AttributeDefinitions": [
            {
                "AttributeName": "LockID",
                "AttributeType": "S"
            }
        ],
        "TableName": "tf-apache-locks",
        "KeySchema": [
            {
                "AttributeName": "LockID",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2024-12-17T14:35:48.078000-03:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 0,
            "WriteCapacityUnits": 0
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:536697232168:table/tf-apache-locks",
        "TableId": "157da151-dd71-4327-879f-b78396685163",
        "BillingModeSummary": {
            "BillingMode": "PAY_PER_REQUEST"
        }
    }
}
```

```
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pin2/terraform-apache 15:12:49 (master 8d395cd) $ aws configure --profile terraform-admin
AWS Access Key ID [****************QKEL]:
AWS Secret Access Key [****************zizc]:
Default region name [None]: us-east-1
Default output format [None]: json
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pin2/terraform-apache 15:13:08 (master 8d395cd) $
```





Evaluar código terraform y credenciales aws:

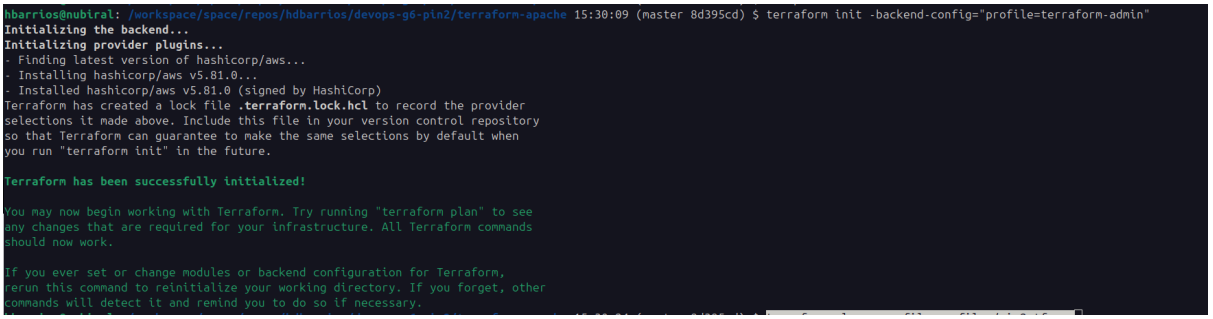
Lista de comandos de Terraform que puedes usar para probar y luego aplicar tu configuración en los archivos de Terraform:

1. Inicializar Terraform

Primero, se debe inicializar el entorno de Terraform. Esto instalará los proveedores necesarios y configurará tu backend (si lo estás usando).

```
terraform init -backend-config="profile=terraform-admin"
```

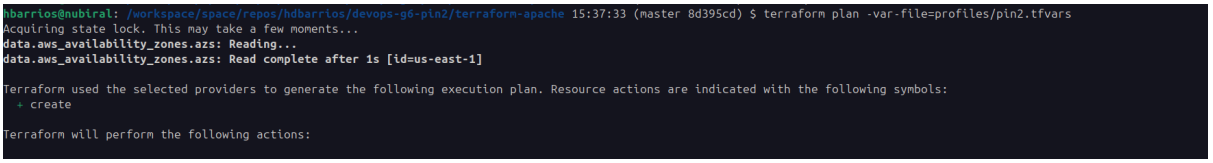
Este comando se ejecuta una sola vez cuando configuras un nuevo proyecto Terraform o cuando haces cambios en los proveedores y módulos.



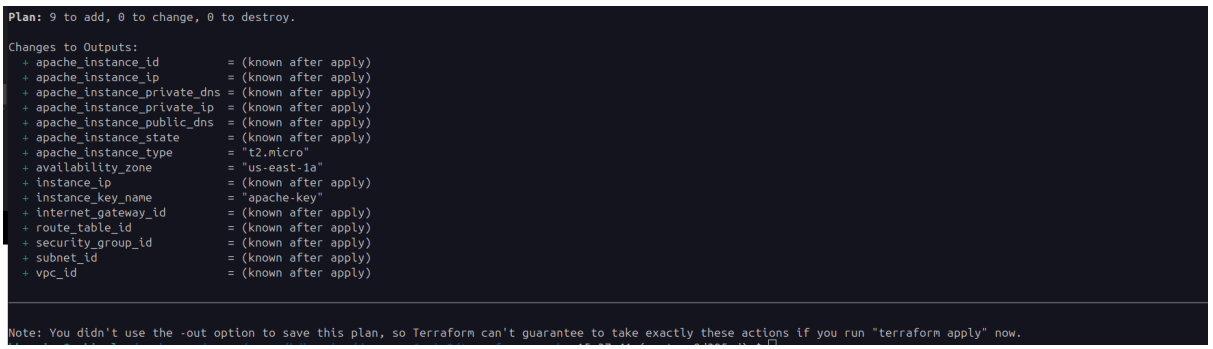
2. Verificar la configuración (Planificación)

Para revisar qué cambios realizará Terraform en tu infraestructura, se puede ejecutar el comando `terraform plan`. Este comando no realizará ningún cambio, solo mostrará una descripción detallada de lo que se va a hacer.

```
terraform plan -var-file=profiles/pin2.tfvars -out=tfplan
```



…



Para este proyecto se debe especificar el archivo de variables `pin2.tfvars` con la opción `-var-file` para que Terraform use las configuraciones definidas en ese archivo.

Salida esperada: Terraform mostrará un resumen de los recursos que se van a crear, modificar o destruir.

Nota:

- se puede ejecutar para validar sintaxis:`terraform validate`

Permite tener código de salida si se quiere implementar CI/CD:

```
terraform plan -var-file=profiles/pin2.tfvars -out=tfplan
-detailed-exitcode && echo $?
```

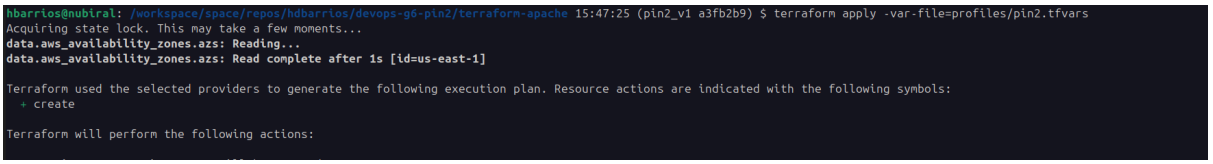
Código de salida 0: No hay cambios
Código de salida 1: Ocurrió un error
Código de salida 2: Se detectaron cambios

3. Aplicar la configuración (Ejecutar cambios)

Si todo está bien con el plan y se puede aplicar los cambios, ejecuta:

```
terraform apply -var-file=profiles/pin2.tfvars
```

Terraform pedirá confirmación antes de proceder. Al estar seguro de que los cambios son correctos, se escribe `yes` para confirmar.

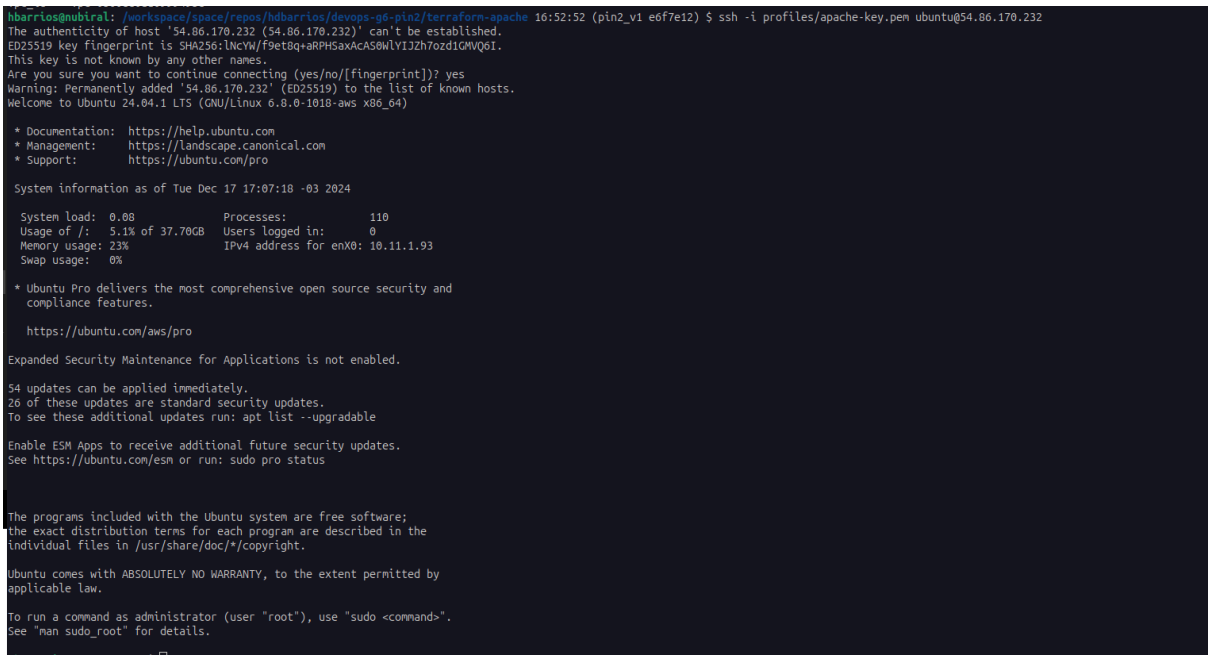


…

```
Plan: 9 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + apache_instance_id          = (known after apply)
  + apache_instance_ip          = (known after apply)
  + apache_instance_private_dns = (known after apply)
  + apache_instance_private_ip  = (known after apply)
  + apache_instance_public_dns  = (known after apply)
  + apache_instance_state       = (known after apply)
  + apache_instance_type        = "t2.micro"
  + availability_zone           = "us-east-1a"
  + instance_ip                 = (known after apply)
  + instance_key_name           = "apache-key"
  + internet_gateway_id         = (known after apply)
  + route_table_id              = (known after apply)
  + security_group_id           = (known after apply)
  + subnet_id                   = (known after apply)
  + vpc_id                      = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

```
  Enter a value: yes

aws_key_pair.apache_key: Creating...
aws_vpc.vpc: Creating...
aws_key_pair.apache_key: Creation complete after 2s [id=apache-key]
aws_vpc.vpc: Still creating... [10s elapsed]
aws_vpc.vpc: Creation complete after 15s [id=vpc-0f9d40af618b4fafa]
aws_internet_gateway.igw: Creating...
aws_route_table.route_table: Creating...
aws_subnet.subnet_public: Creating...
aws_security_group.sg: Creating...
aws_internet_gateway.igw: Creation complete after 1s [id=igw-09f19e228215208c3]
aws_route_table.route_table: Creation complete after 2s [id=rtb-0e5009baced2cf730]
aws_route.route: Creating...
aws_route.route: Creation complete after 1s [id=r-rtb-0e5009baced2cf7301080289494]
aws_security_group.sg: Creation complete after 5s [id=sg-01068faf4d07ded5a]
aws_subnet.subnet_public: Still creating... [10s elapsed]
aws_subnet.subnet_public: Creation complete after 12s [id=subnet-040b27741e9732ab6]
aws_route_table_association.subnet_association_public: Creating...
aws_instance.apache_server: Creating...
aws_route_table_association.subnet_association_public: Creation complete after 2s [id=rtbassoc-06efd632373e8986b]
aws_instance.apache_server: Still creating... [10s elapsed]
aws_instance.apache_server: Creation complete after 15s [id=i-0cc7c34358d149039]
Releasing state lock. This may take a few moments...

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.
```

```
Outputs:

apache_instance_id = "i-09402c46f21e474d8"
apache_instance_ip = "54.86.170.232"
apache_instance_private_dns = "ip-10-11-1-93.ec2.internal"
apache_instance_private_ip = "10.11.1.93"
apache_instance_public_dns = "ec2-54-86-170-232.compute-1.amazonaws.com"
apache_instance_state = "running"
apache_instance_type = "t2.micro"
availability_zone = "us-east-1a"
instance_ip = "54.86.170.232"
instance_key_name = "apache-key"
internet_gateway_id = "igw-0bf33492afe92fa21"
route_table_id = "rtb-0134c2c0b302752ac"
security_group_id = "sg-05c31ceef628a23dc"
subnet_id = "subnet-036017f22ad2969e7"
vpc_id = "vpc-0560dd0aa099949bd"
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pin2/terraform-apache 16:52:52 (pin2_v1 e6f7e12) $
```

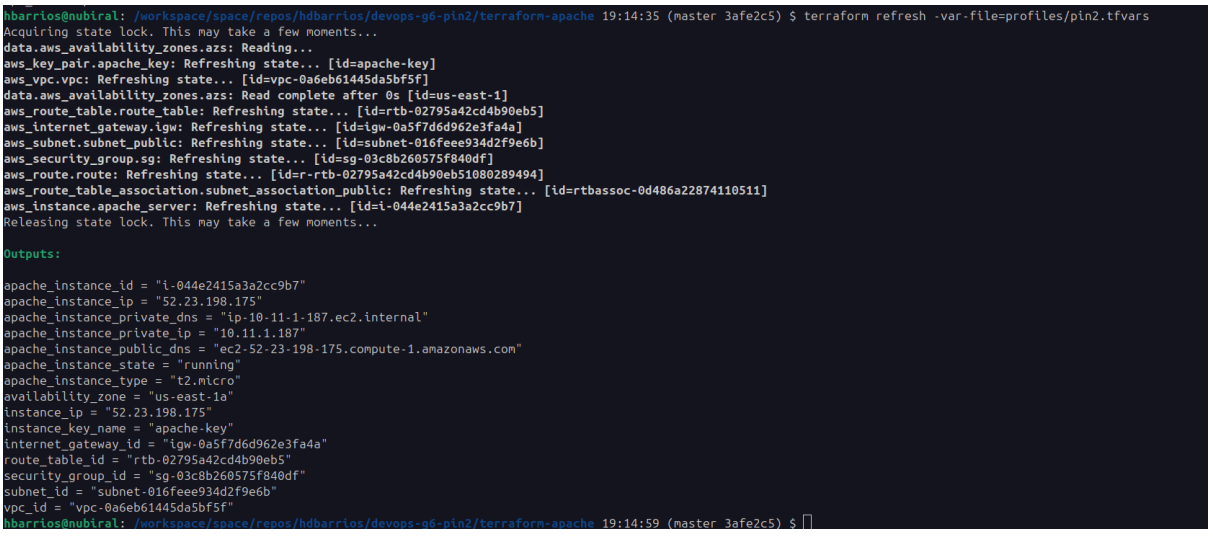


4. Ver los resultados de la aplicación

Después de aplicar la configuración, Terraform mostrará la salida definida en tu archivo `outputs.tf`, si has configurado algún bloque `output`.

Para obtener información adicional sobre los recursos creados (por ejemplo, la dirección IP pública de una instancia EC2), se puede usar:

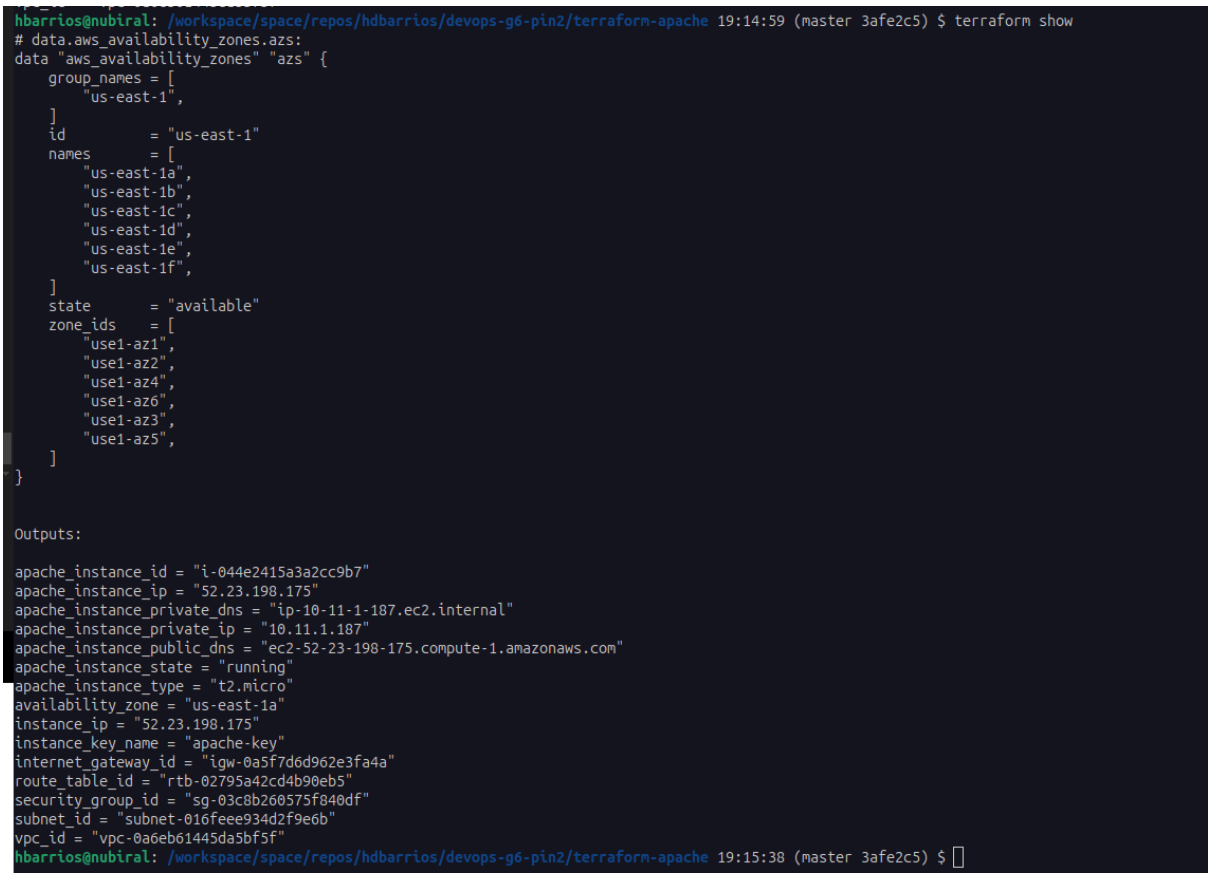
```
terraform output -var-file=profiles/pin2.tfvar
```



5. Comprobar el estado actual de Terraform

Si se requiere ver el estado actual de la infraestructura gestionada por Terraform, ejecutar:

```
terraform show
```



Salida en yml

(

instalación:

```
export VERSION=v4.2.0 && export BINARY=yq_linux_amd64 && wget https://github.com/mikefarah/yq/releases/download/${VERSION}/${BINARY}.tar.gz -O - | tar xz && sudo mv ${BINARY} /usr/bin/yq
```

)

```
terraform show -json | jq . | yq eval -P
```

6. Destruir la infraestructura (opcional)

Si se necesita destruir todos los recursos que has creado (por ejemplo, para probar la limpieza), puedes usar:

```
terraform destroy -var-file=profiles/pin2.tfvars
```

Terraform solicitara confirmación. Escribe `yes` para proceder.

```
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pin2/terraform-apache 16:00:03 (pin2_v1 a3fb2b9) $ terraform destroy -var-file=profiles/pin2.tfvars
Acquiring state lock. This may take a few moments...
data.aws_availability_zones.azs: Reading...
aws_key_pair.apache_key: Refreshing state... [id=apache-key]
aws_vpc.vpc: Refreshing state... [id=vpc-0141420a8020eb0ab]
data.aws_availability_zones.azs: Read complete after 0s [id=us-east-1]
aws_internet_gateway.igw: Refreshing state... [id=igw-0f7ad5d8608604952]
aws_route_table.route_table: Refreshing state... [id=rtb-0c75dee3bf37de848]
aws_subnet.subnet_public: Refreshing state... [id=subnet-0d5a7331fc3305f2e]
aws_security_group.sg: Refreshing state... [id=sg-0225273c5153035b3]
aws_route.route: Refreshing state... [id=r-rtb-0c75dee3bf37de8481080289494]
aws_route_table_association.subnet_association_public: Refreshing state... [id=rtbassoc-0d18cd54d3b1d7728]
aws_instance.apache_server: Refreshing state... [id=i-037e4e3c229a4113b]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.apache_server will be destroyed
  - resource "aws_instance" "apache_server" {
      - ami                                  = "ami-0e2c8caa4b6378d8c" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:536697232168:instance/i-037e4e3c229a4113b" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-1a" -> null
      - cpu_core_count                       = 1 -> null
```

...

```
Plan: 0 to add, 0 to change, 9 to destroy.

Changes to Outputs:
  - apache_instance_id          = "i-037e4e3c229a4113b" -> null
  - apache_instance_ip          = "3.88.201.241" -> null
  - apache_instance_private_dns = "ip-10-11-1-12.ec2.internal" -> null
  - apache_instance_private_ip  = "10.11.1.12" -> null
  - apache_instance_public_dns  = "ec2-3-88-201-241.compute-1.amazonaws.com" -> null
  - apache_instance_state       = "running" -> null
  - apache_instance_type        = "t2.micro" -> null
  - availability_zone           = "us-east-1a" -> null
  - instance_ip                 = "3.88.201.241" -> null
  - instance_key_name           = "apache-key" -> null
  - internet_gateway_id         = "igw-0f7ad5d8608604952" -> null
  - route_table_id              = "rtb-0c75dee3bf37de848" -> null
  - security_group_id           = "sg-0225273c5153035b3" -> null
  - subnet_id                   = "subnet-0d5a7331fc3305f2e" -> null
  - vpc_id                      = "vpc-0141420a8020eb0ab" -> null

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes
```

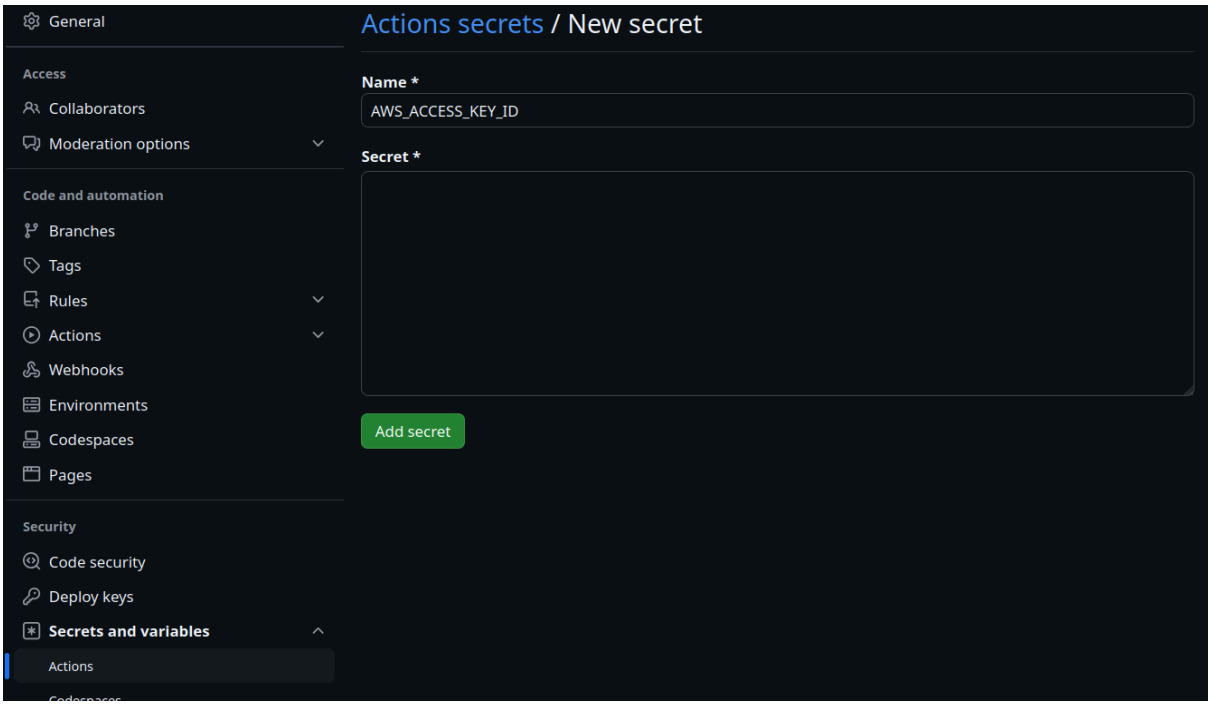
...

```
  Enter a value: yes

aws_route_table_association.subnet_association_public: Destroying... [id=rtbassoc-06efd632373e8986b]
aws_route.route: Destroying... [id=r-rtb-0e5009baced2cf7301080289494]
aws_instance.apache_server: Destroying... [id=i-0cc7c34358d149039]
aws_route_table_association.subnet_association_public: Destruction complete after 2s
aws_route.route: Destruction complete after 2s
aws_internet_gateway.igw: Destroying... [id=igw-09f19e228215208c3]
aws_route_table.route_table: Destroying... [id=rtb-0e5009baced2cf730]
aws_route_table.route_table: Destruction complete after 2s
aws_instance.apache_server: Still destroying... [id=i-0cc7c34358d149039, 10s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-09f19e228215208c3, 10s elapsed]
aws_instance.apache_server: Still destroying... [id=i-0cc7c34358d149039, 20s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-09f19e228215208c3, 20s elapsed]
aws_instance.apache_server: Still destroying... [id=i-0cc7c34358d149039, 30s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-09f19e228215208c3, 30s elapsed]
aws_instance.apache_server: Still destroying... [id=i-0cc7c34358d149039, 40s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-09f19e228215208c3, 40s elapsed]
aws_instance.apache_server: Still destroying... [id=i-0cc7c34358d149039, 50s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-09f19e228215208c3, 50s elapsed]
aws_instance.apache_server: Still destroying... [id=i-0cc7c34358d149039, 1m0s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-09f19e228215208c3, 1m0s elapsed]
aws_internet_gateway.igw: Destruction complete after 1m1s
aws_instance.apache_server: Destruction complete after 1m6s
aws_key_pair.apache_key: Destroying... [id=apache-key]
aws_subnet.subnet_public: Destroying... [id=subnet-040b27741e9732ab6]
aws_security_group.sg: Destroying... [id=sg-01068faf4d07ded5a]
aws_key_pair.apache_key: Destruction complete after 0s
aws_subnet.subnet_public: Destruction complete after 1s
aws_security_group.sg: Destruction complete after 2s
aws_vpc.vpc: Destroying... [id=vpc-0f9d40af618b4fafa]
aws_vpc.vpc: Destruction complete after 1s

Destroy complete! Resources: 9 destroyed.
hbarrios@nubiral: /workspace/space/repos/hdbarrios/devops-g6-pin2/terraform-apache 16:12:49 (pin2_v1 e6f7e12) $
```

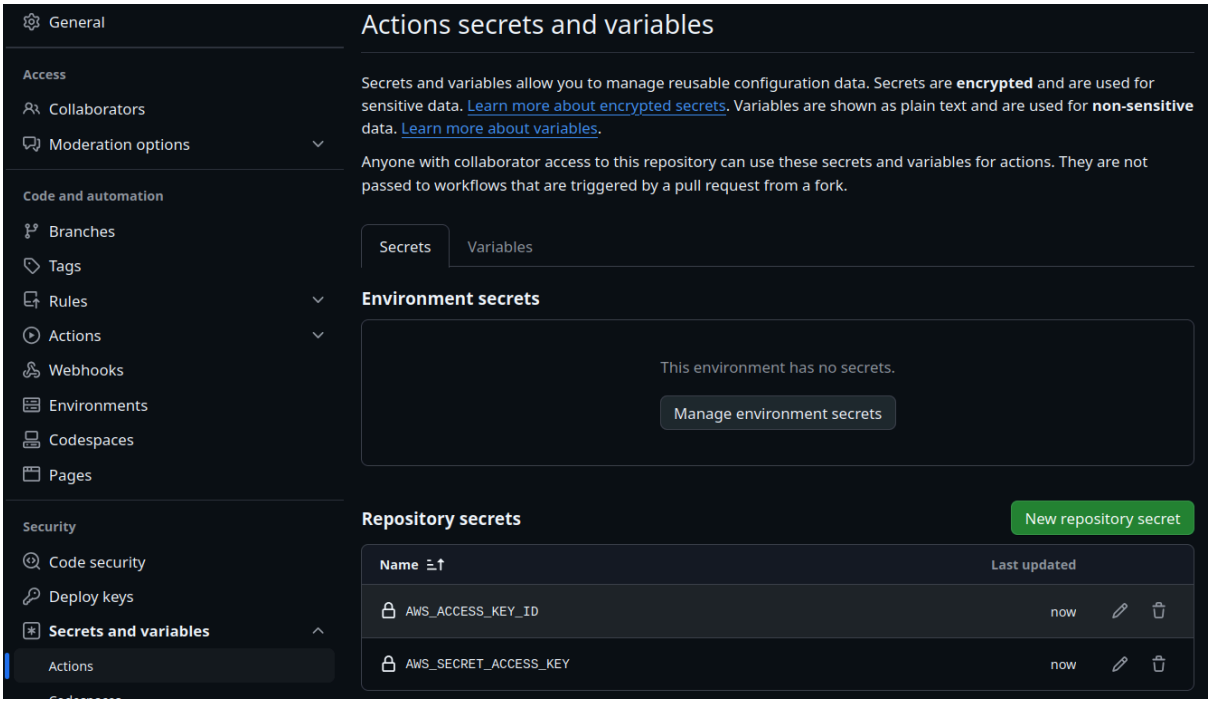
GitHub Actions:

Se procede a configurar los secrets de acceso a la cuenta AWS, los nombres deben de coincidir luego en los workflows de las actions de github que se diseñaron.

AWS_ACCESS_KEY_ID



AWS_SECRET_ACCESS_KEY



MundosE - *https://mundose.com/index.html*

Los Workflows diseñados fueron:

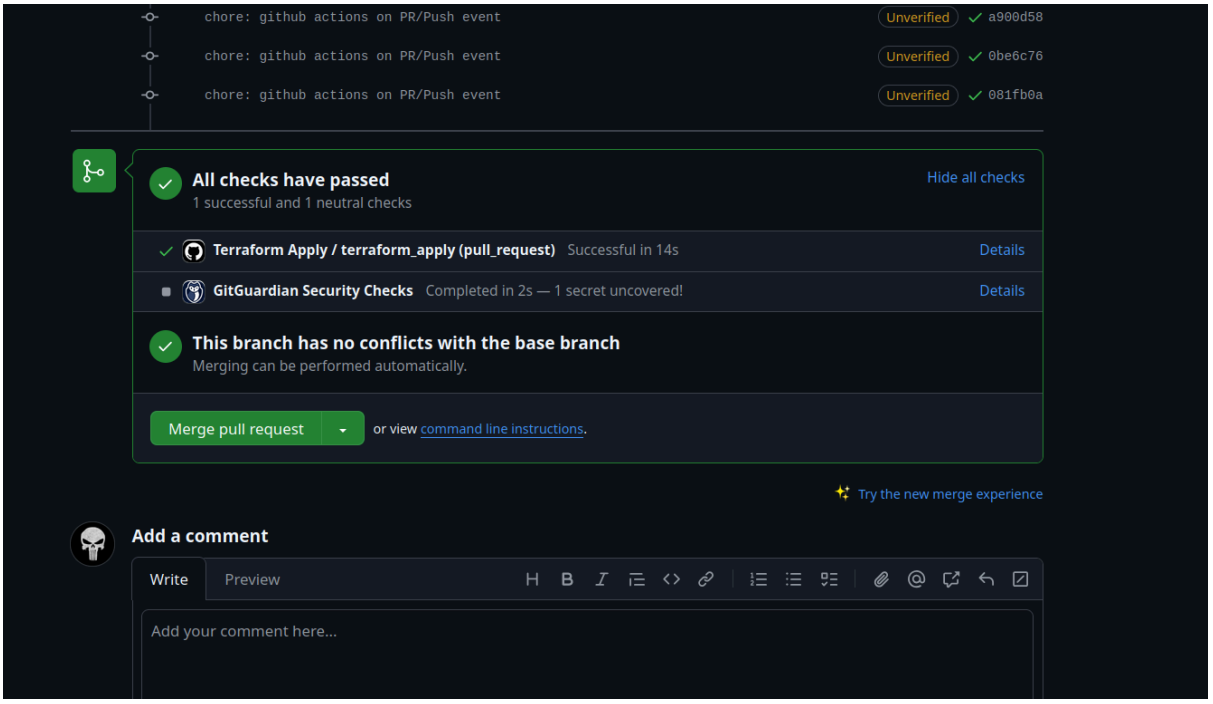
Terraform Apply, solo se ejecuta en condiciones;

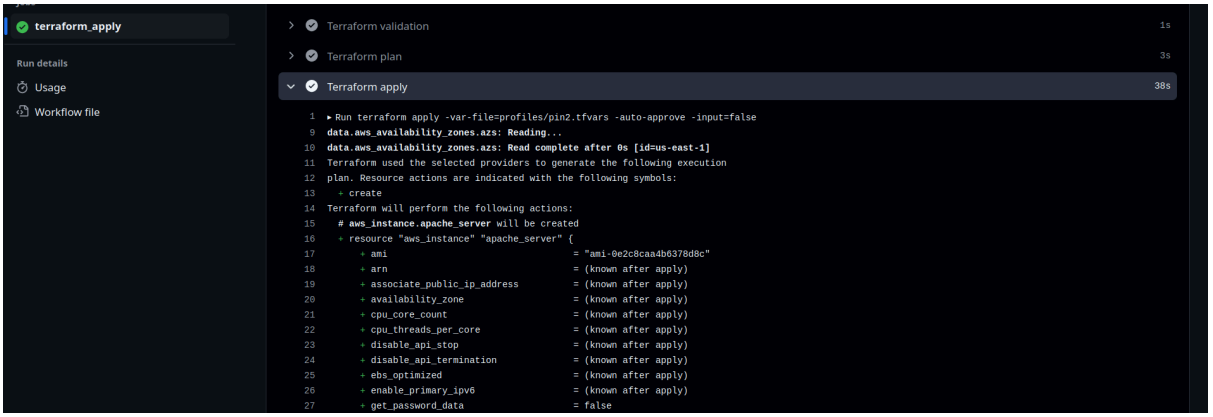
```
on:
  push:
    branches:
      - master
  pull_request:
    branches:
      - master
  workflow_dispatch:
    branches:
      - master
```

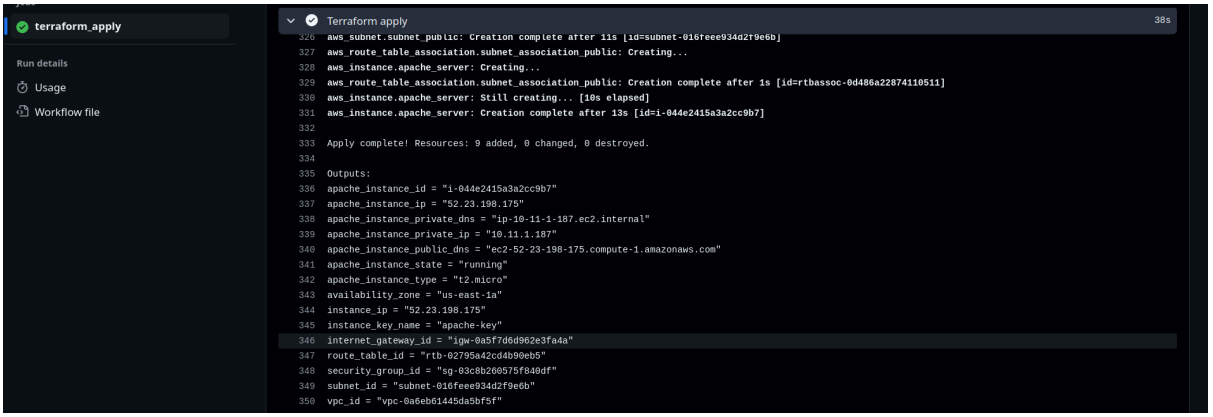
Terraform Destroy, solo se ejecuta con la condicion;

```
on:
  workflow_dispatch:
    branches:
      - master
```

Ejemplo de ejecución con PR:







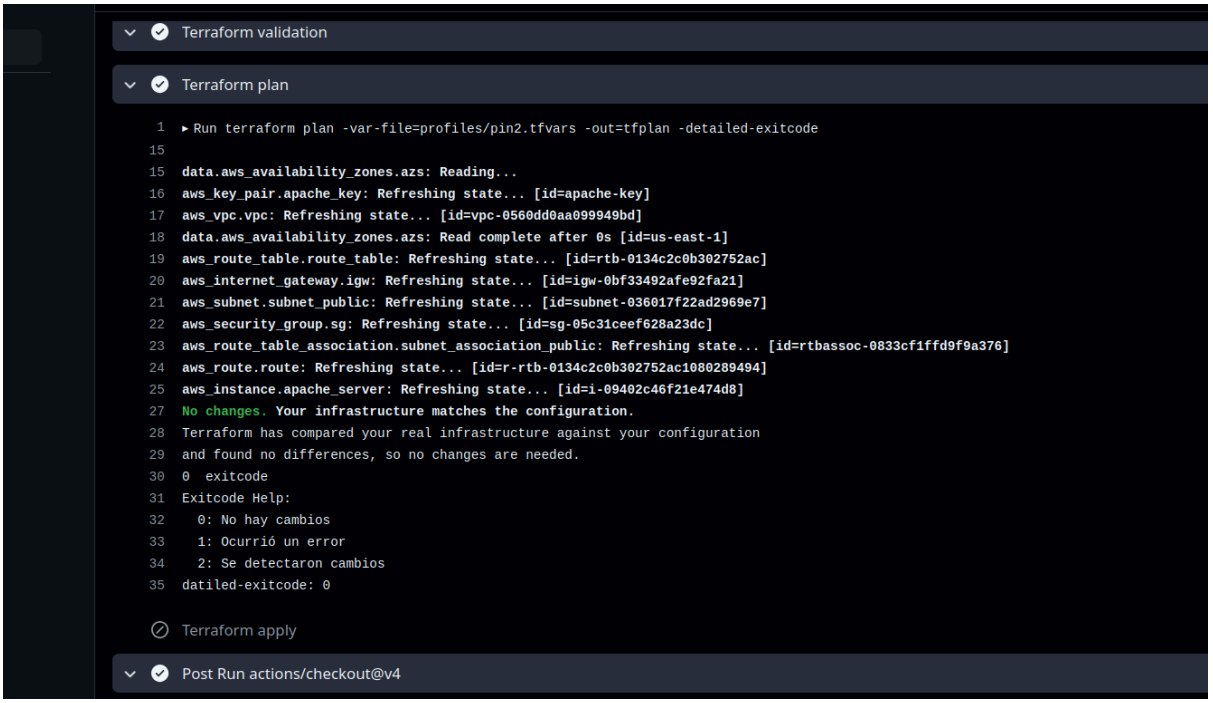
Si todo sale bien, se podrá observar la IP pública de la instancia y si la usamos para navegar en un browser podremos ver la ejecución del apache:

edu.mundose.com - PIN2

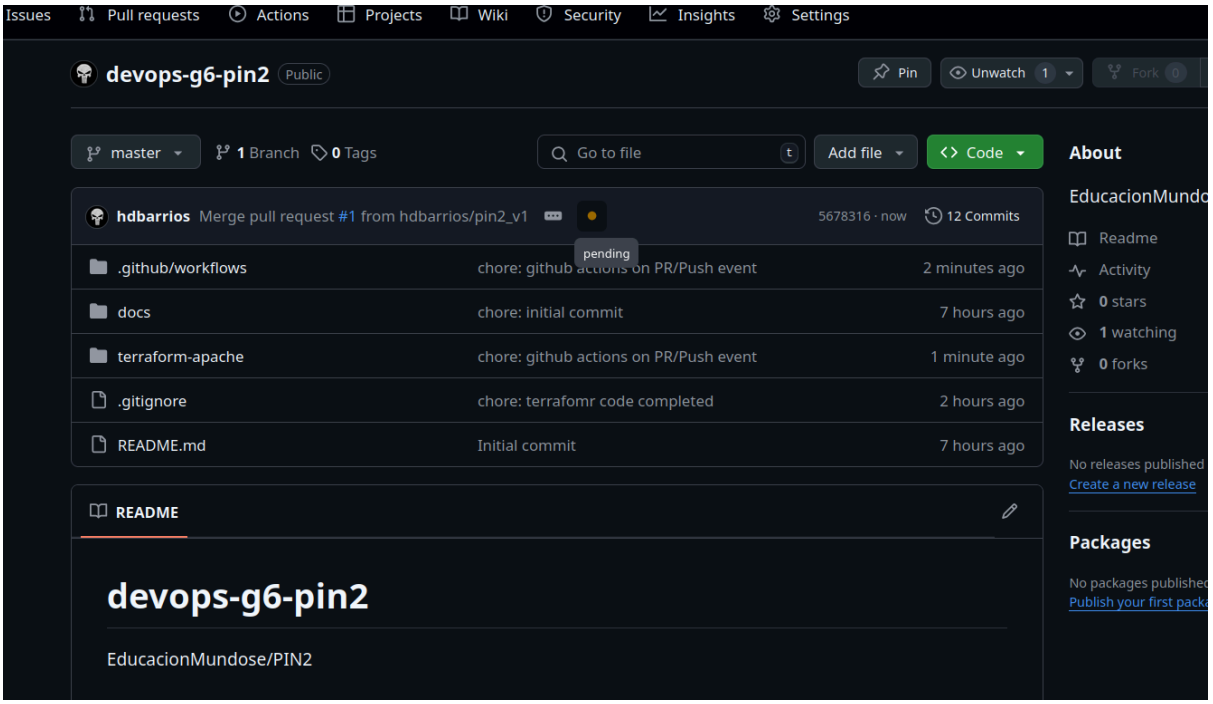
Integrantes	Email
Juan Pablo Heyda	juanpabloh.123@gmail.com
Renzo Emiliano Carletti	renzocarletti@hotmail.com / pipito1498@gmail.com
Johanna Dominguez	johisd9@hotmail.com
Lucas Bufano	lucas.bufano2@gmail.com
Hector Barrios	hdbarrios@gmail.com

Si se ejecuta de nuevo el workflow Terraform Apply indicará que no existen cambios por aplicar.

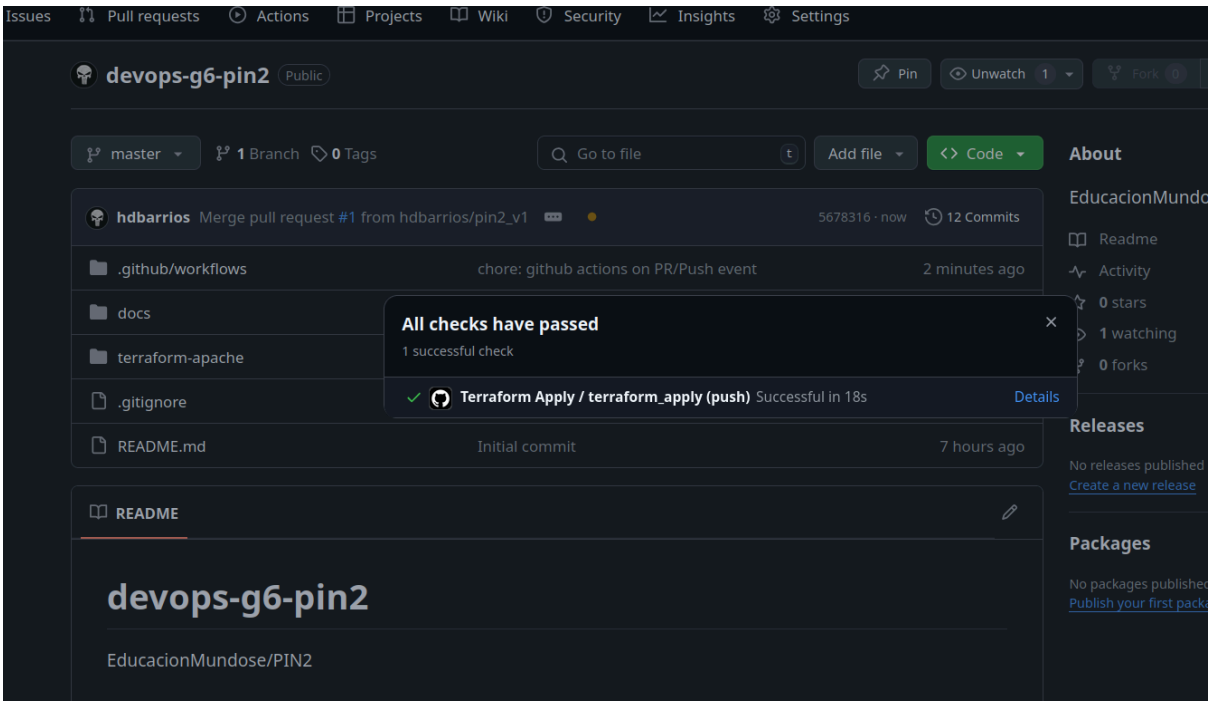
MundosE - *https://mundose.com/index.html*



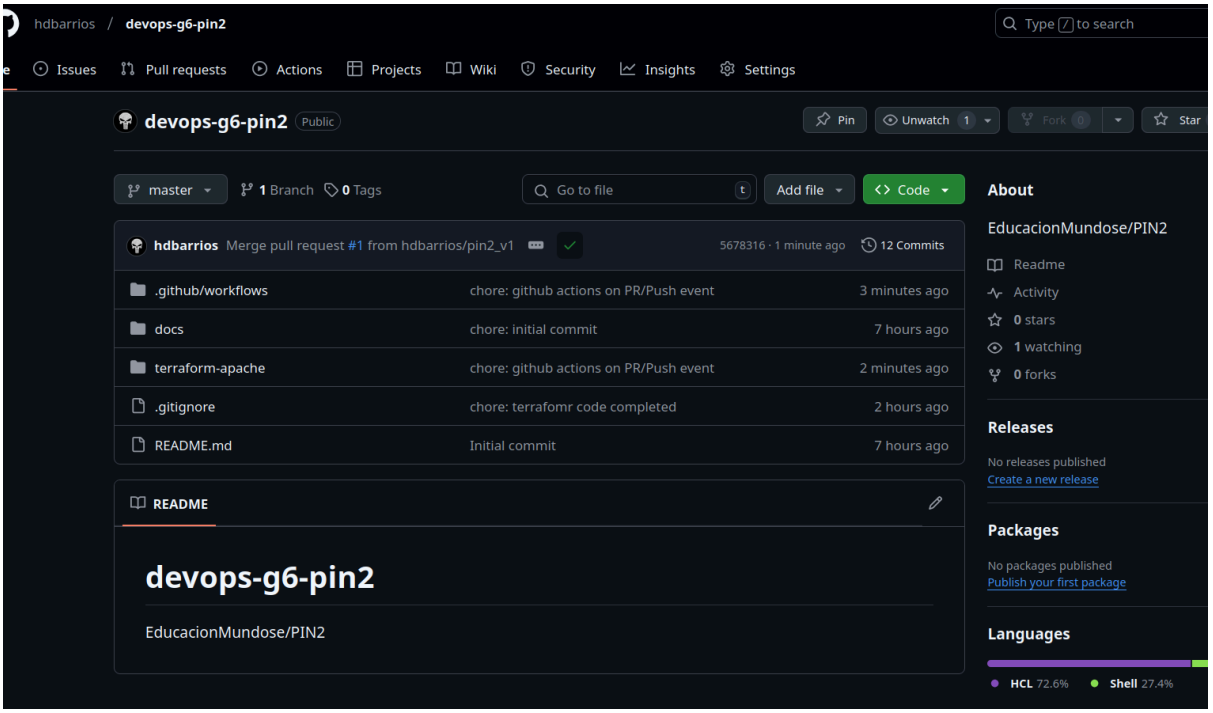
Estatus del Workflow sobre el commit-merge:



MundosE - *https://mundose.com/index.html*

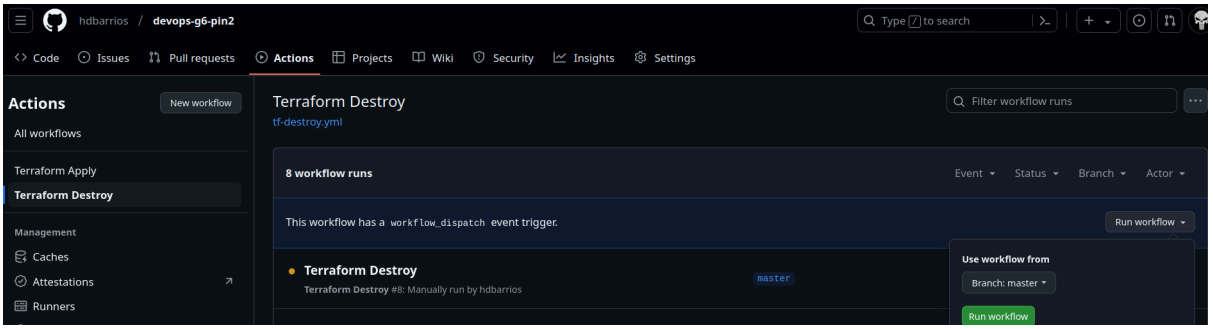


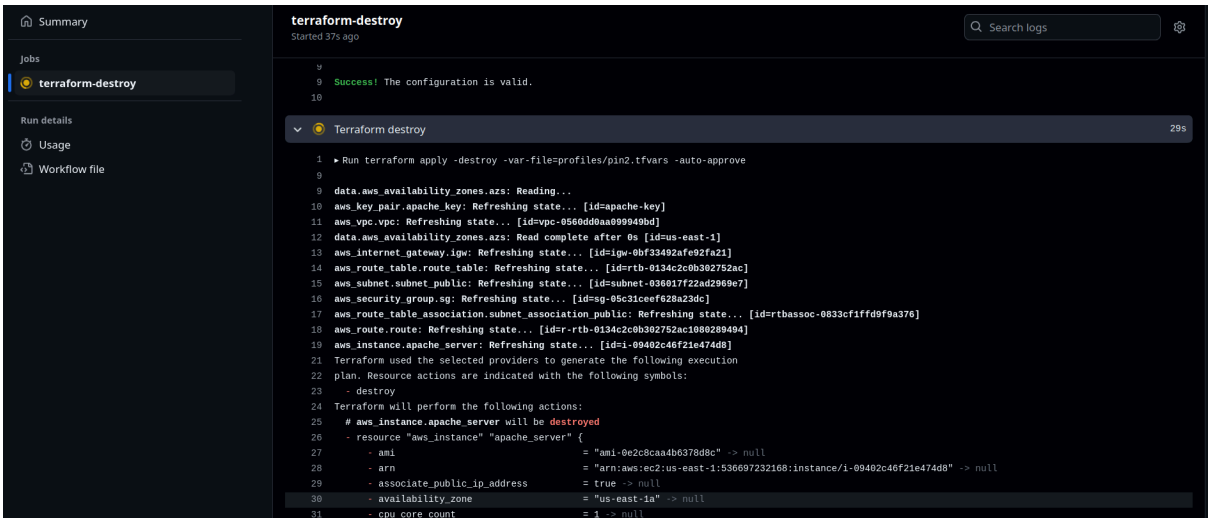
Al finalizar se observa failed o successful dependiendo el estado del job, para el ejemplo “Sucessful”



MundosE - *https://mundose.com/index.html*

El Workflow Terraform Destroy solo se ejecutara on deman:





...

MundosE - *https://mundose.com/index.html*