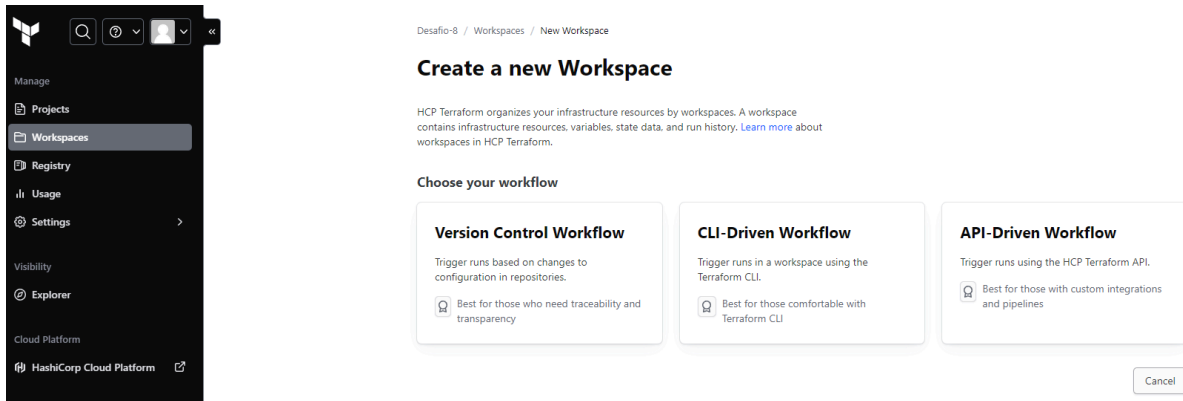


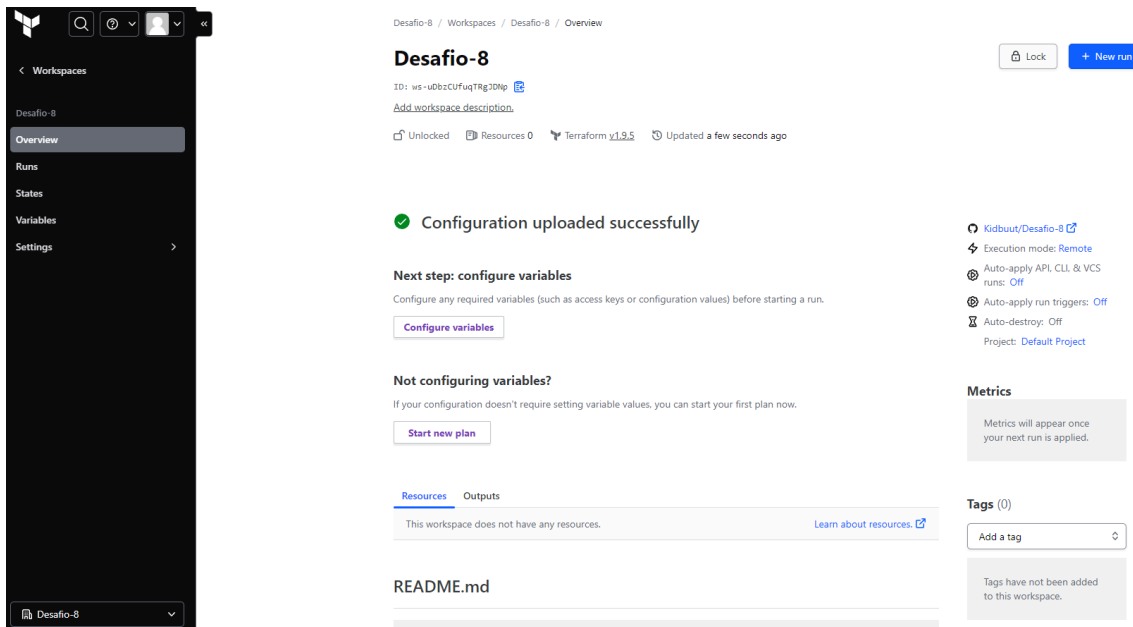
## Desafio 8

Para iniciar este desafío tenemos que tener cuentas e instalación en nuestro entorno local las herramientas:

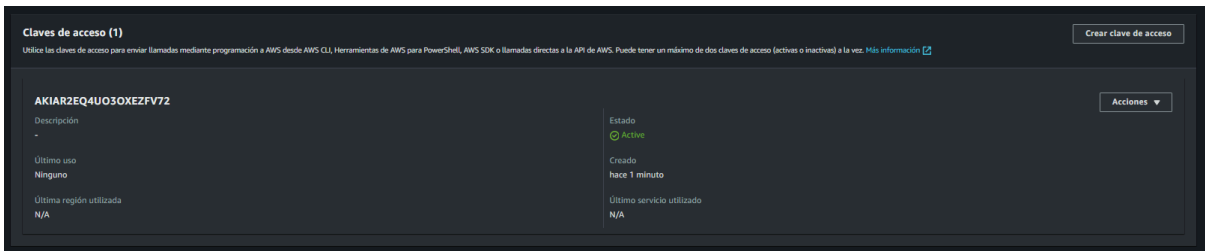
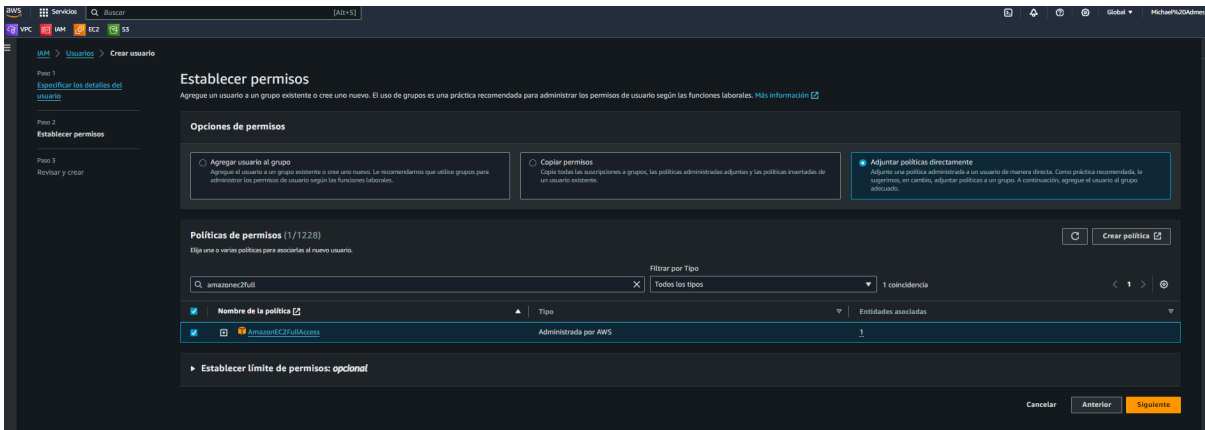
- Terraform
- Github
- AWS



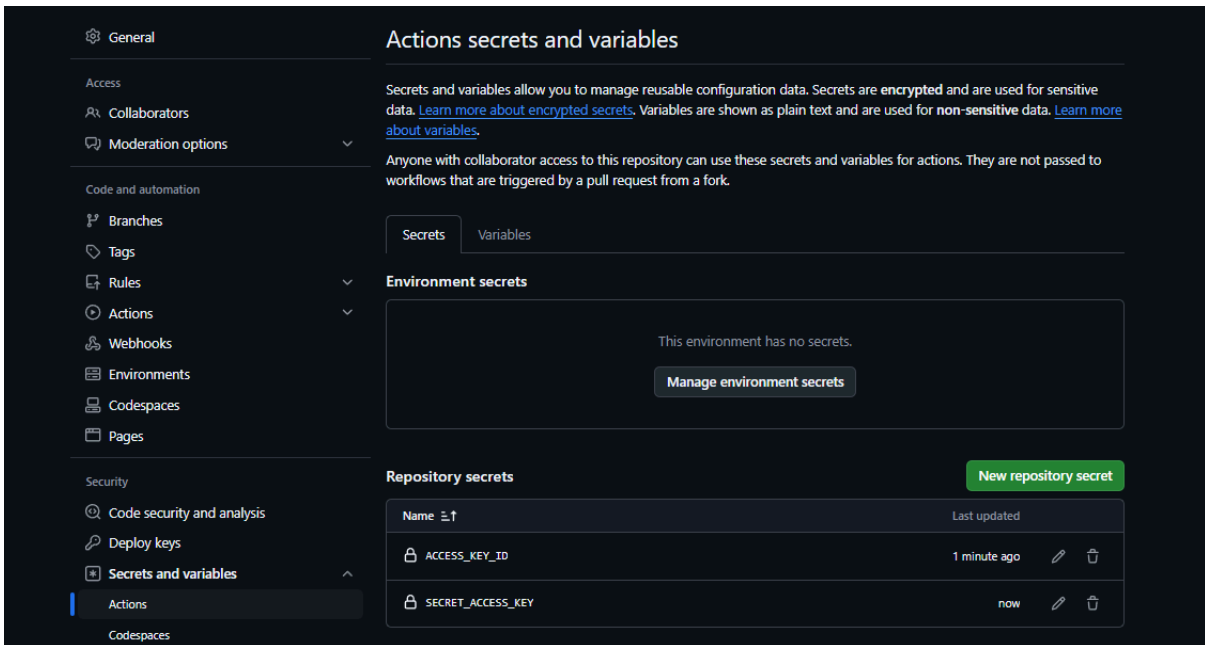
Una vez teniendo todo seteado pasamos a la dirección <https://app.terraform.io/> a la creación de nuestra organización en mi caso **Desafio-8** seguidamente del Workspaces con la configuración para que tenga acceso a nuestro Github.



Luego seguimos con la creación de nuestro usuario en AWS, y le damos los permisos correspondientes y seguidamente generamos la Clave de acceso.



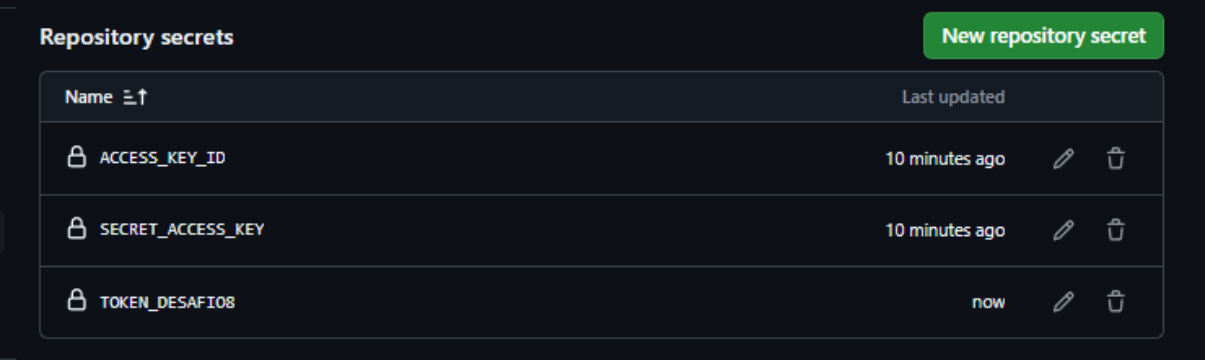
Una vez creado el usuario nos dirigimos a Github a la ventana de Settings-Credential and variables-action y agregamos las credenciales del usuario de AWS.



Luego de esto en AWS creamos una Api Token y lo hacemos desde EC2-Setting-Token



Esa clave la pegamos en las credenciales de Github junto con las que agregamos anteriormente.

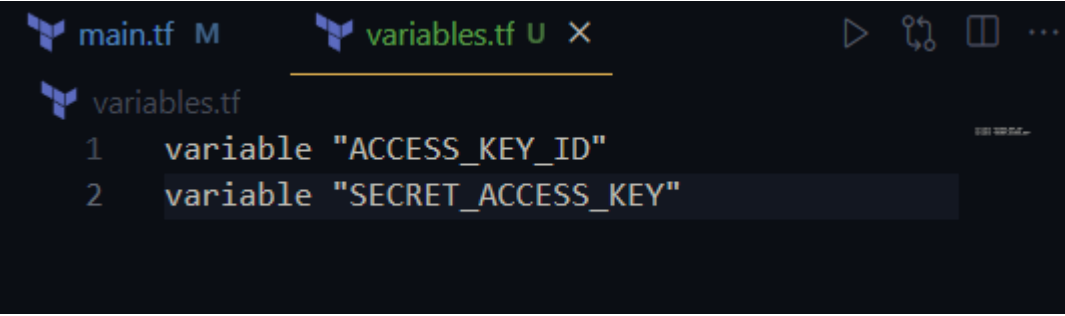


Repository secrets		New repository secret	
Name ↕	Last updated		
ACCESS_KEY_ID	10 minutes ago		
SECRET_ACCESS_KEY	10 minutes ago		
TOKEN_DESAFIO8	now		

Seguimos con la creación del código de Terraform

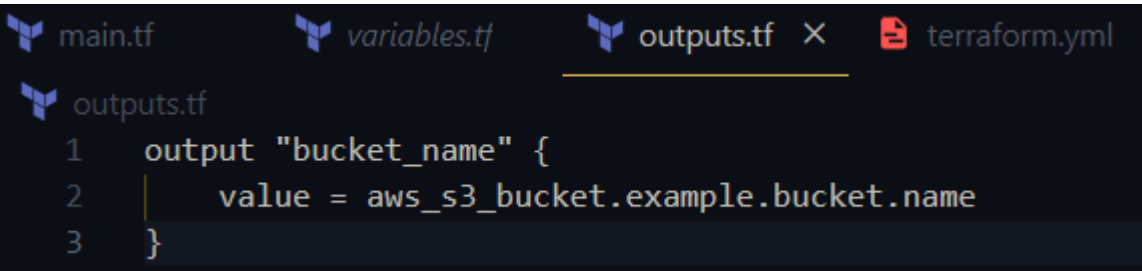
1 - archivo main.tf

2 - archivo variables.tf



```
main.tf M  variables.tf U X
variables.tf
1 variable "ACCESS_KEY_ID"
2 variable "SECRET_ACCESS_KEY"
```

3 - outputs.tf

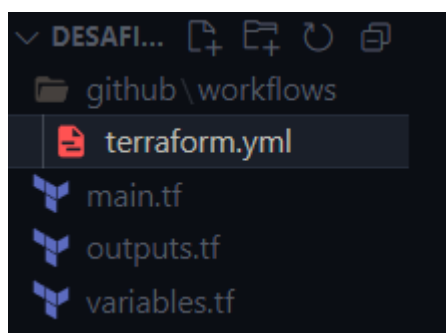


```
main.tf  variables.tf  outputs.tf X  terraform.yml
outputs.tf
1 output "bucket_name" {
2   value = aws_s3_bucket.example.bucket.name
3 }
```

#### 4 - terraform.yml

```
main.tf  variables.tf  outputs.tf  terraform.yml X
github > workflows > terraform.yml
1  name: Provision t3.micro EC2
2
3  on:
4    push:
5      branches:
6        - '**' # Se ejecutará en cualquier rama con cualquier commit
7      workflow_dispatch:
8      inputs:
9        ec2-name:
10         description: EC2 name
11         required: true
12         default: 'App Server'
13         type: string
14
15  jobs:
16    provision-ec2:
17      runs-on: ubuntu-latest
18      steps:
19        - uses: actions/checkout@v3
20
21        - uses: actions/setup-node@v3
22          with:
23            node-version: '14'
24
25        - name: Configure AWS credentials
26          uses: aws-actions/configure-aws-credentials@v1
27          with:
28            aws-access-key-id: '${{ secrets.AWS_ACCESS_KEY_ID }}'
29            aws-secret-access-key: '${{ secrets.AWS_SECRET_ACCESS_KEY }}'
30            aws-region: us-east-2
31
32        - name: Setup Terraform
33          uses: hashicorp/setup-terraform@v2
34          with:
35            terraform_wrapper: false
36
37        - name: Terraform Apply
38          id: apply
39          env:
```

nuestro código quedaría así



una vez terminado de configurar todo y crear nuestro código en nuestra terminal corremos los siguientes códigos:

## Terraform init

```
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.67.0...
- Installed hashicorp/aws v5.67.0 (self-signed, key ID 34365D9472D7468F)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/plugins/signing.html

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, we recommend adding version constraints in a required_providers block
in your configuration, with the constraint strings suggested below.

* hashicorp/aws: version = "~> 5.67.0"

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.
```

## Terraform validate

```
C:\Users\Michael\Desktop\Desafio-8>terraform validate
Success! The configuration is valid.
```

Despues de muchos intentos me dio infinidades de errores el cual no me corrio porque me decia que tenia un error con la VPC

```
    }
+ id                = (known after apply)
+ ingress           = [
+   {
+     cidr_blocks = [
+       "0.0.0.0/0",
+     ]
+     description = ""
+     from_port   = 0
+     ipv6_cidr_blocks = []
+     prefix_list_ids = []
+     protocol     = "tcp"
+     security_groups = []
+     self         = false
+     to_port      = 65535
+   },
+ ]
+ name              = "app-sg"
+ name_prefix       = (known after apply)
+ owner_id          = (known after apply)
+ revoke_rules_on_delete = false
+ tags              = {
+   "Name" = "app-sg"
+ }
+ tags_all          = {
+   "Name" = "app-sg"
+ }
+ vpc_id            = (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  instance_id           = (known after apply)
  instance_private_ip   = (known after apply)
  instance_public_ip    = (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: yes

aws_security_group.app-sg: Creating...
Error: creating Security Group (app-sg): operation error EC2: CreateSecurityGroup, https response error StatusCode: 400, RequestID: df942881-8c98-4f2f-a2b4-adc267a5f50a, api error VPCIdNotSpecified: No default VPC for this user

on main.tf line 25, in resource "aws_security_group" "app-sg":
25: resource "aws_security_group" "app-sg" {
```

busque informacion y ayuda por todos lados y no encuentre nada, dejo capturas de cómo todo estaba bien configurado y me seguia dando el error.

```
17
18  resource "aws_security_group" "app_sg" {
19      vpc_id = "vpc-01d9d403a6d7a6b84" # Reemplaza con tu VPC ID
20
```

VPC > Sus VPC > vpc-01d9d403a6d7a6b84

vpc-01d9d403a6d7a6b84 / Desafío-VPC

Acciones

Detalles

Información

ID de la VPC vpc-01d9d403a6d7a6b84	Estado Available	Nombres de host de DNS Desactivado	Resolución de DNS Habilitado
Tenencia Default	Conjunto de opciones de DHCP dopt-02325e7022184ddb1	Tabla de enrutamiento principal rtb-09440df38b9afcd96	ACL de red principal acl-0cb4dd49f5be4dbde
VPC predeterminada No	CIDR IPv4 10.0.0.0/16	Grupo IPv6 -	CIDR IPv6 -
Métricas de uso de direcciones de red Desactivado	Grupos de reglas del firewall de DNS de Route 53 Resolver	ID de propietario 124857918390	

Current Run

update

CURRENT

✖ Errored

#run-T6MetmR7rHYxkpUr | Kidbuut triggered via GitHub | Branch main | 1dcf4e7

4 minutes ago

Run List

All 8

Needs Attention 0

Errored 8

Running 0

On Hold 0

Success 0

Search Runs

Q

Status

Operation

Source

update

CURRENT

✖ Errored

#run-T6MetmR7rHYxkpUr | Kidbuut triggered via GitHub | Branch main | 1dcf4e7

4 minutes ago

update

✖ Errored

#run-wJZ3oZDaLRYaWNwR | Kidbuut triggered via GitHub | Branch main | a5d147e

10 minutes ago

update

✖ Errored

#run-89BTCCCeutzrwiV9 | Kidbuut triggered via GitHub | Branch main | 5691882

12 minutes ago

update

✖ Errored

#run-Jezz67toadG1idHn | Kidbuut triggered via GitHub | Branch main | f6879f8

15 minutes ago

update

✖ Errored

#run-NnxFkHCTnNho8BY7 | Kidbuut triggered via GitHub | Branch main | 093bede

17 minutes ago

update

✖ Errored

#run-GMBvD7sVgJoYLyAh | Kidbuut triggered via GitHub | Branch main | 1396697

an hour ago

update

✖ Errored

#run-dBKx56Ehaqcaczn | Kidbuut triqqered via GitHub | Branch main | 50a0c2e

an hour ago

el próximo desafío lo haré mejor.