LABORATORIO: CREAR UNA INSTANCIA EC2 EN AWS CON TERRAFORM

GIAN LUCAS FIGUEROA FELIZOLA

DOCENTE

SALOMON SEGUNDO DE LA HOZ

ELECTIVA

CLOUD COMPUTING

FACULTAD DE INGENIERIA Y TECNOLOGIAS

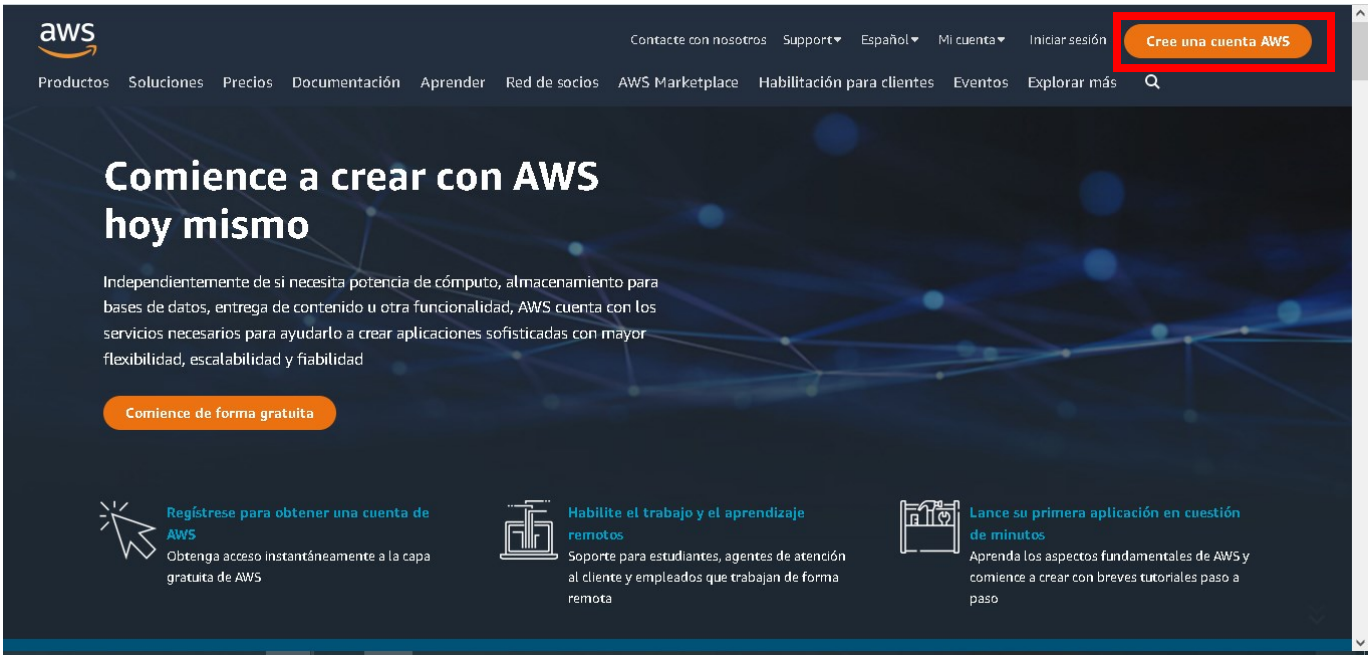
UNIVERSIDAD POPULAR DEL CESAR

VALLEDUPAR-CESAR

2022

LABORATORIO: COMO CREAR UNA UNA INSTACIA EC2 EN AWS CON TERRAFORM

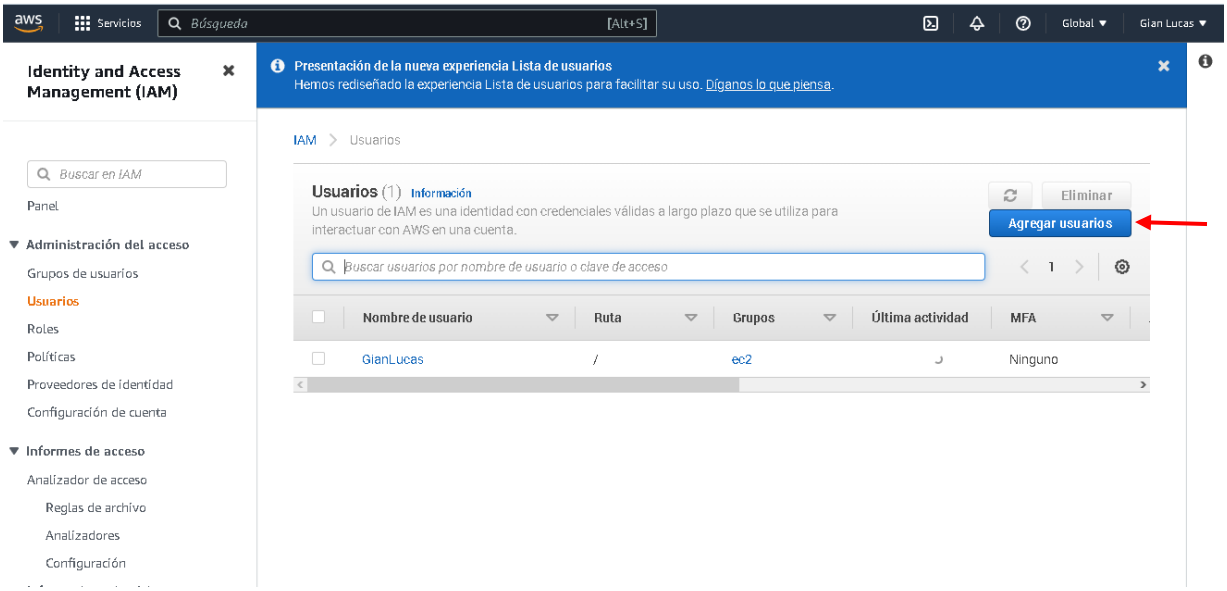
Lo primero que se debe hacer es crear nuestra cuenta AWS



Acá seguimos todos los pasos para el registro, necesitaras un tarjeta para poder crear la cuenta.

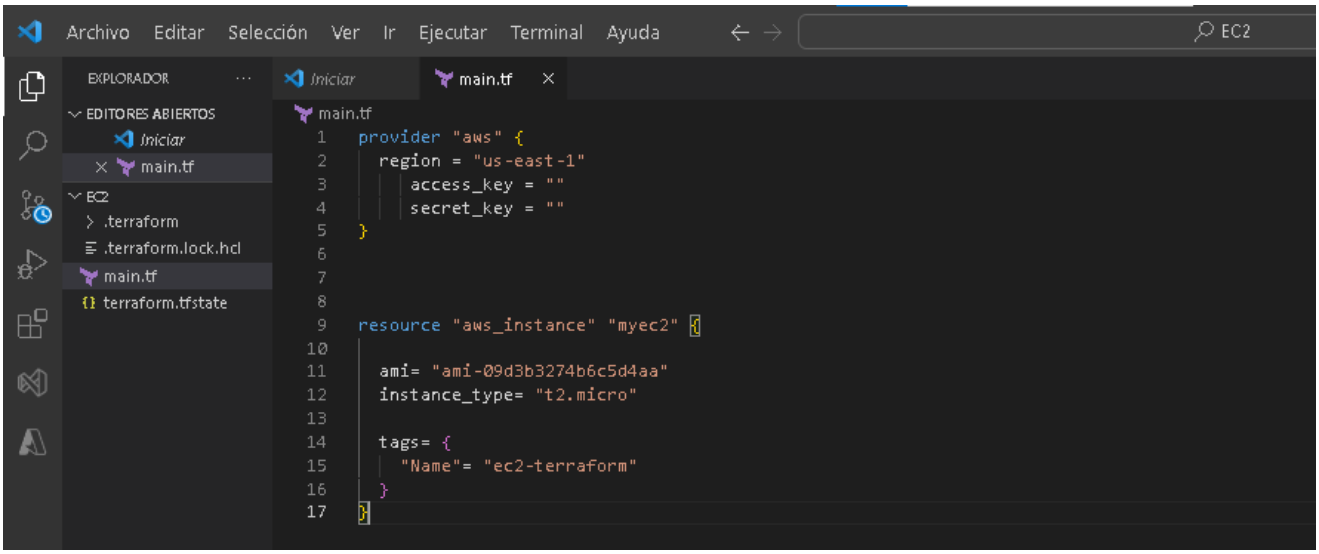
Para poder crear la instancia, se necesitará crear un usuario, el cual nos proporcionará las claves de acceso para crear la instancia.

Este lo creamos en el Identity and Access Management (IAM), en el apartado de usuario



En este caso tengo mi usuario ya creado con las políticas necesarias, para crear uno nuevo se le da en el botón **Agregar Usuarios.**

Al finalizar la creación del usuario nos proporciona las claves de acceso las cual debemos guardar porque de lo contrario deberemos crear unas nuevas.



Seguidamente se crea un archivo **.tf** para escribir el código que nos ayudara a la creación de esta instancia.

Guardamos y Abrimos la terminal, en la que escribiremos comando **terraform init**

```
C:\Users\Gian Lucas\Documents\EC2>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.39.0...
- Installed hashicorp/aws v4.39.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
```

Vemos que ha inicializado exitosamente

Ahora escribimos **terraform plan**

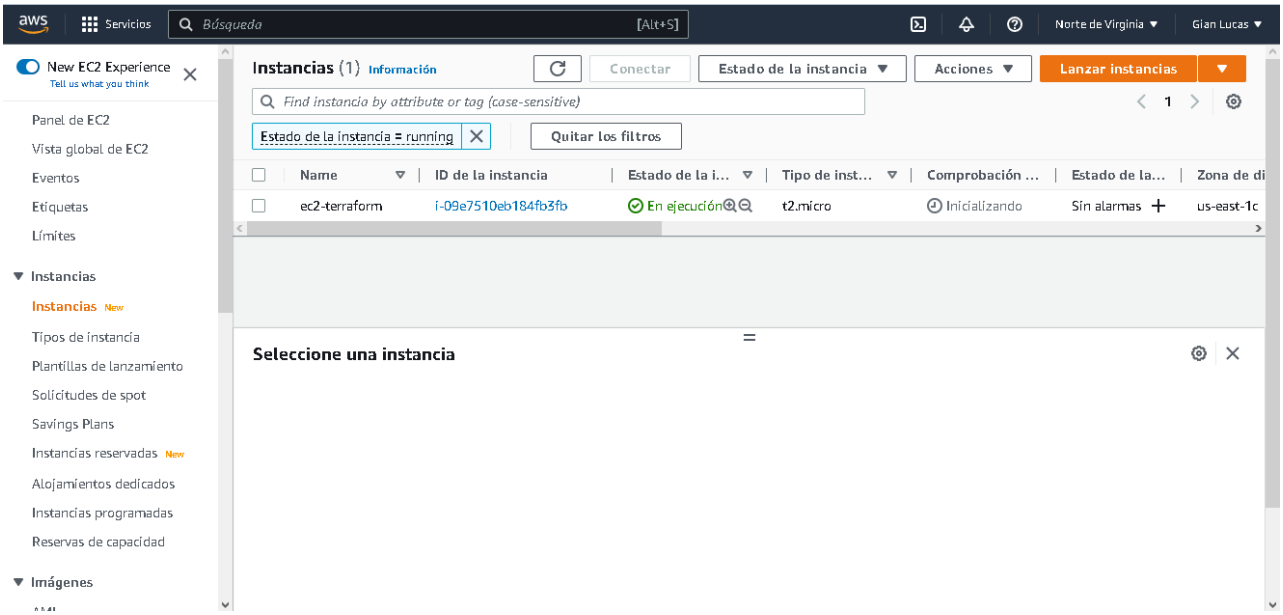
```
C:\Users\Gian Lucas\Documents\EC2>terraform plan

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

Terraform will perform the following actions:

  # aws_instance.myec2 will be created
  + resource "aws_instance" "myec2" {
      + ami                                  = "ami-09d3b3274b6c5d4aa"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
```

Ahora escribimos **terraform apply** para que no los aplique y cree la instancia.

```
C:\Users\Gian Lucas\Documents\EC2>terraform apply

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

Terraform will perform the following actions:

  # aws_instance.myec2 will be created
  + resource "aws_instance" "myec2" {
      + ami                                  = "ami-09d3b3274b6c5d4aa"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
```

```
Plan: 1 to add, 0 to change, 0 to destroy.

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_instance.myec2: Creating...
aws_instance.myec2: Still creating... [10s elapsed]
aws_instance.myec2: Still creating... [20s elapsed]
aws_instance.myec2: Still creating... [30s elapsed]
aws_instance.myec2: Creation complete after 32s [id=i-09e7510eb184fb3fb]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\Users\Gian Lucas\Documents\EC2>
```

Cuando nos salga *Enter a value* escribimos **yes**



Nos vamos al panel de EC2 y buscamos instancias, y vemos que la que creamos esta ya arriba y en ejecución.

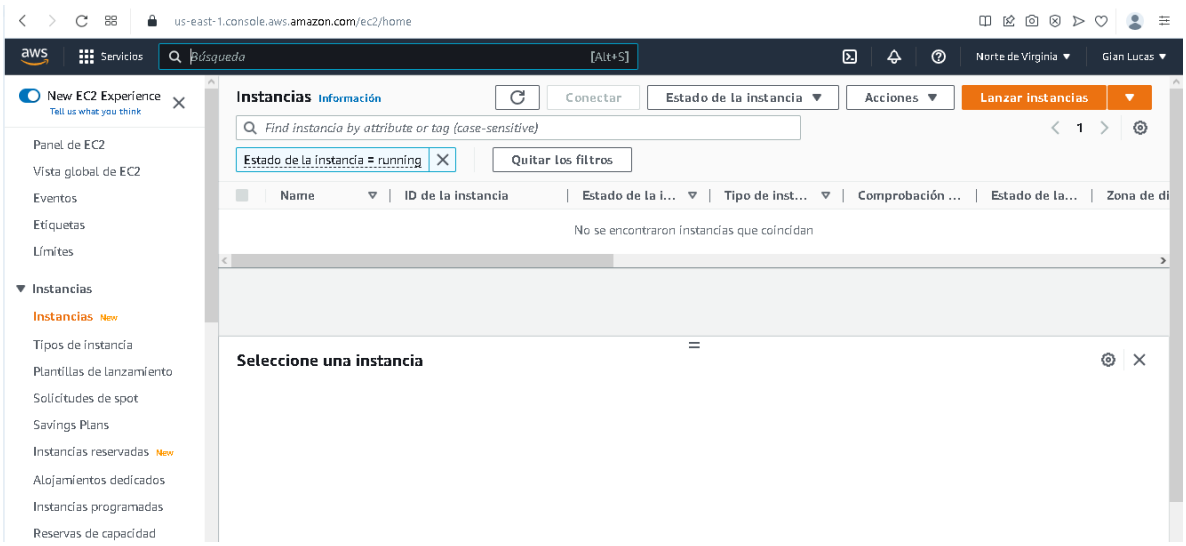
Por ultimo, borramos la instancia con el comando **terraform destroy**

Nos saldrá denuevo para que aceptemos realizar ese cambio y escribimos yes

```
C:\Users\Gian Lucas\Documents\EC2>terraform destroy
aws_instance.myec2: Refreshing state... [id=i-09e7510eb184fb3fb]

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.myec2 will be destroyed
  - resource "aws_instance" "myec2" {
      - ami                                  = "ami-09d3b3274b6c5d4aa" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:721849456432:instance/i-09e7510eb184fb3fb" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-1c" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-09e7510eb184fb3fb" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
```

```
Destroy complete! Resources: 1 destroyed.
```

Ya por último verificamos nuestra instancia si ha sido borrada



Y podemos observar que ya no nos aparece.

Para la realización de este laboratorio se usaron fuentes diferentes a las suministradas por el docente, las cuales se encontraran los enlaces en la bibliografía.

BIBILIOGRAFIA

https://www.youtube.com/watch?v=8C7LD4Fnh90

https://www.youtube.com/watch?v=IgtAH2ynhus

https://www.youtube.com/watch?v=IexieD7HSQQ