

Crear una instancia EC2 en AWS con Terraform

Estudiante:

Johan Calixto Ramírez Acosta

Docente:

Salomón Segundo de la hoz

Electiva:

Cloud Computing

Facultad de Ingenierías y Tecnológicas

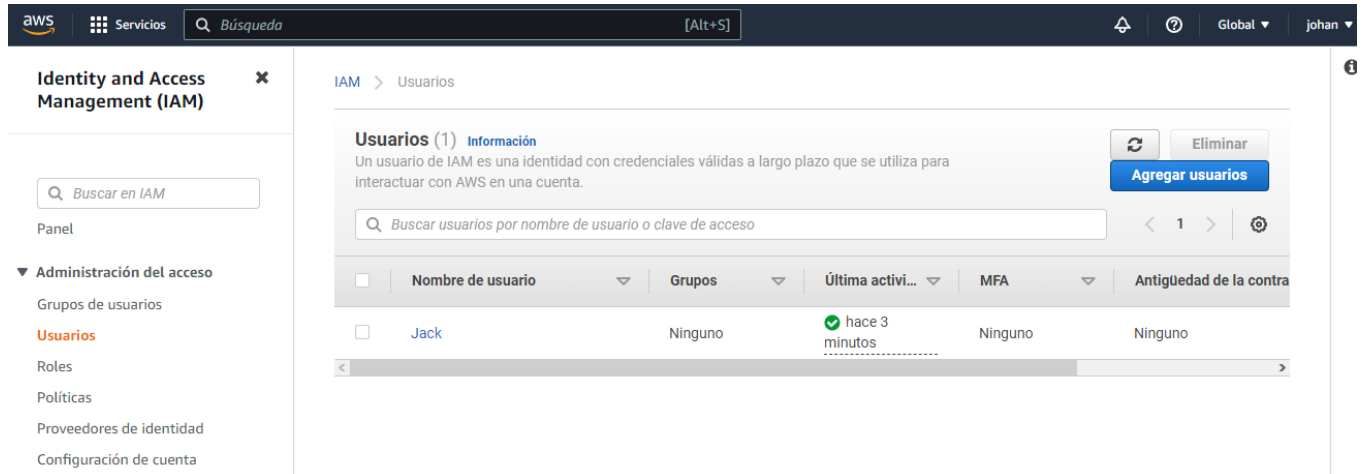
Universidad Popular del Cesar

Valledupar – Cesar

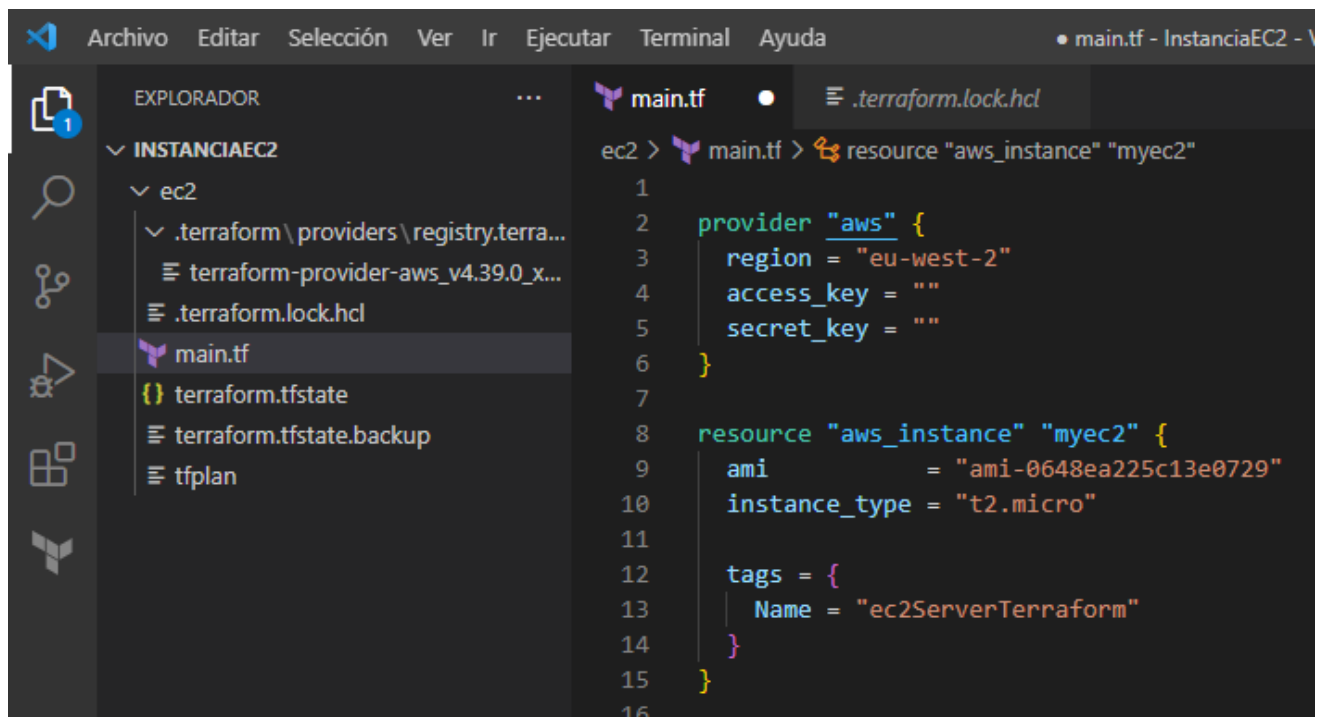
2022

Crear una instancia EC2 en AWS con Terraform

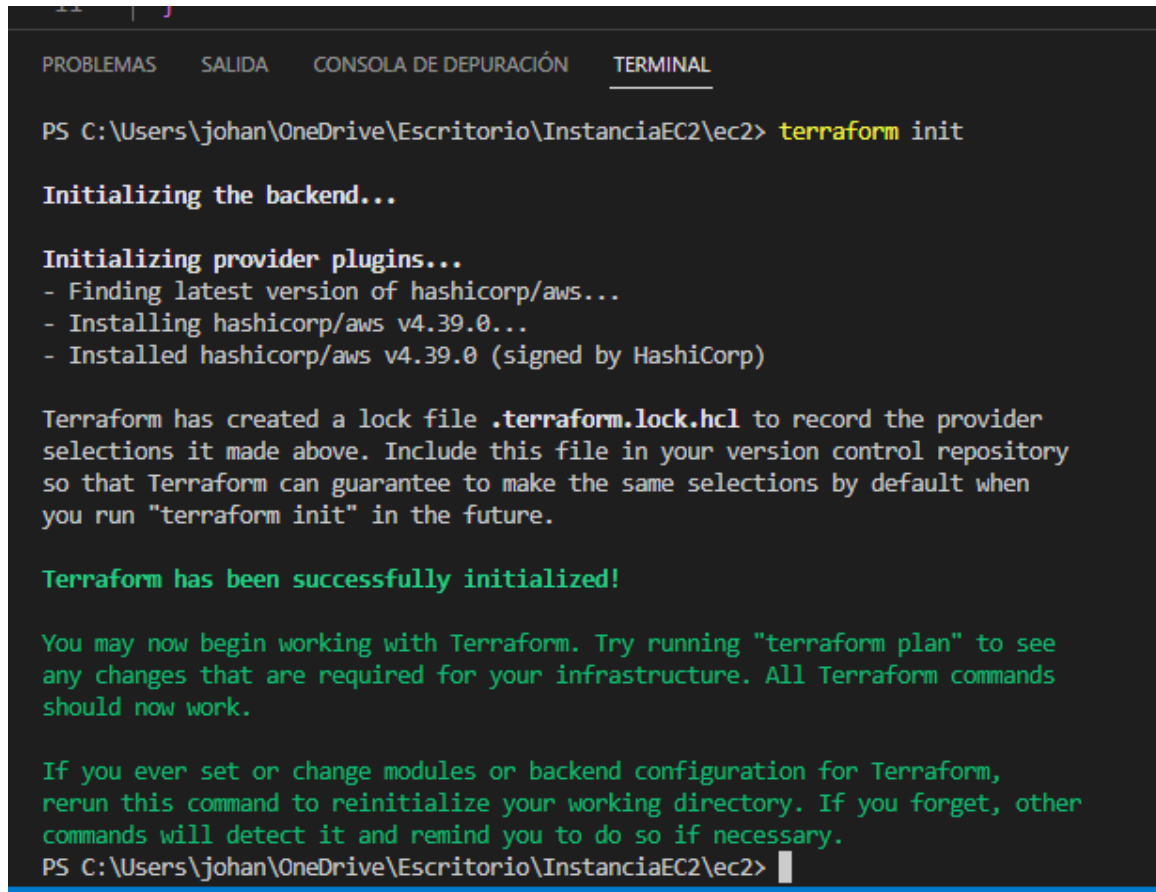
En AWS es importante crear un usuario en Identity and Access Management (IAM) con esto tendremos el access_key y secret_key



Escribimos el siguiente código en Visual Code



Inicializamos el directorio de trabajo que contiene la configuración de Terraform



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\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.
PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\ec2> 
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\ec2> terraform plan -out=tfplan

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-0648ea225c13e0729"
  + 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)
}
```

Por último, aplicamos y creamos la instancia EC2

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

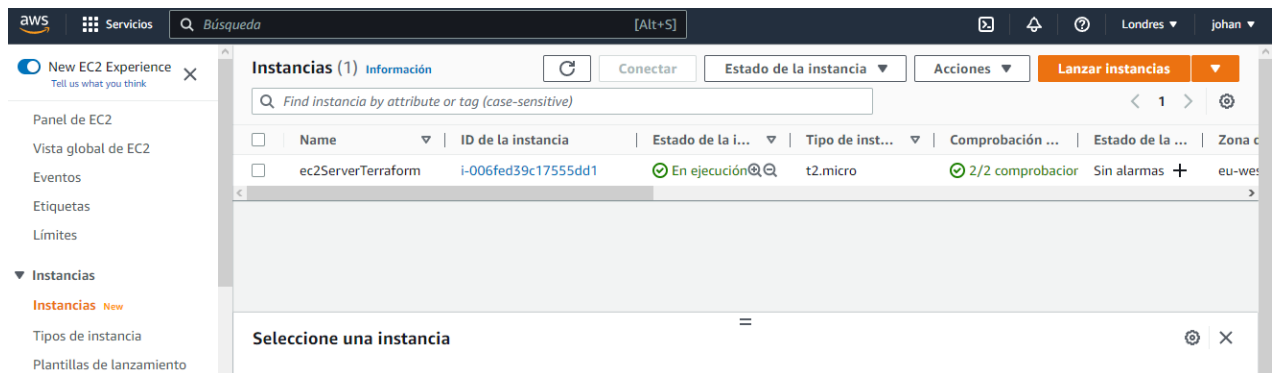
Plan: 1 to add, 0 to change, 0 to destroy.

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
  terraform apply "tfplan"
PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\ec2> terraform apply --auto-approve tfplan
aws_instance.myec2: Creating...
aws_instance.myec2: Still creating... [10s elapsed]
aws_instance.myec2: Still creating... [20s elapsed]
aws_instance.myec2: Creation complete after 23s [id=i-006fed39c17555dd1]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\ec2> █
```

Instancia creada



Ahora borraremos la instancia con Terraform

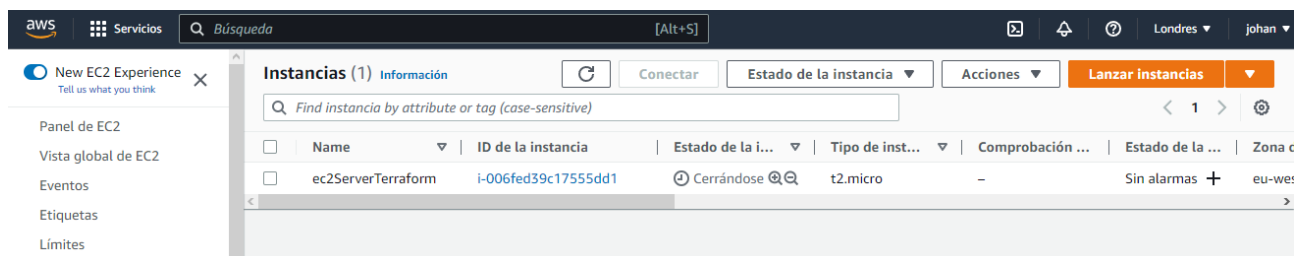
```
PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\ec2> terraform destroy
aws_instance.myec2: Refreshing state... [id=i-006fed39c17555dd1]

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:

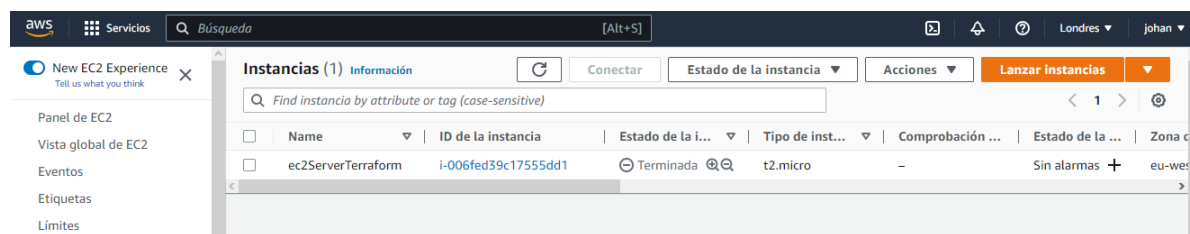
# aws_instance.myec2 will be destroyed
- resource "aws_instance" "myec2" {
  - ami                  = "ami-0648ea225c13e0729" -> null
  - arn                  = "arn:aws:ec2:eu-west-2:476786095040:instance/i-006fed39c17555dd1" -> null
  - associate_public_ip_address = true -> null
  - availability_zone      = "eu-west-2a" -> 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
}
```

Como podemos observar la instancia se detuvo



Finalmente se eliminó la instancia

```
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  
  
aws_instance.myec2: Destroying... [id=i-006fed39c17555dd1]  
aws_instance.myec2: Still destroying... [id=i-006fed39c17555dd1, 10s elapsed]  
aws_instance.myec2: Still destroying... [id=i-006fed39c17555dd1, 20s elapsed]  
aws_instance.myec2: Still destroying... [id=i-006fed39c17555dd1, 30s elapsed]  
aws_instance.myec2: Destruction complete after 31s  
  
Destroy complete! Resources: 1 destroyed.  
PS C:\Users\johan\OneDrive\Escritorio\InstanciaEC2\ec2>
```



Bibliografía

<https://www.youtube.com/watch?v=8C7LD4Fnh90&t=142s>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html#cli-configure-quickstart-creds>