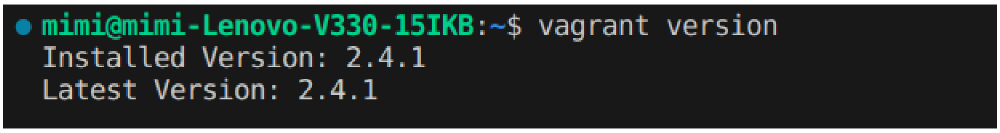
Desafio 11: Miriam Romitelli

| ***Etapa 1: Vagrant*** |

Referencia instalación Vagrant: https://www.redeszone.net/tutoriales/servidores/vagrant-instalacion-configuracion-ejemplos/

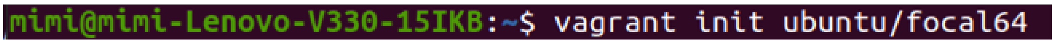
Comando utilizado para instalar el Vagrant: `sudo apt install vagrant`



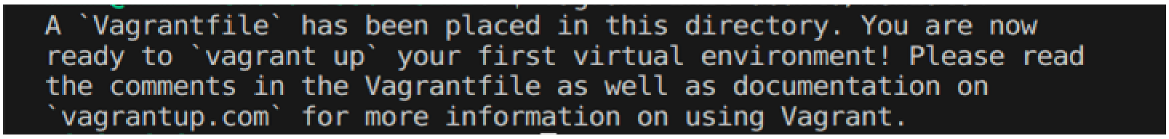
```
● mimi@mimi-Lenovo-V330-15IKB:~$ vagrant version
  Installed Version: 2.4.1
  Latest Version: 2.4.1
```

Crear el Vagrant File: vagrant init ubuntu/focal64

Estoy indicando a *Vagrant* que quiero utilizar la imagen ubuntu/focal64

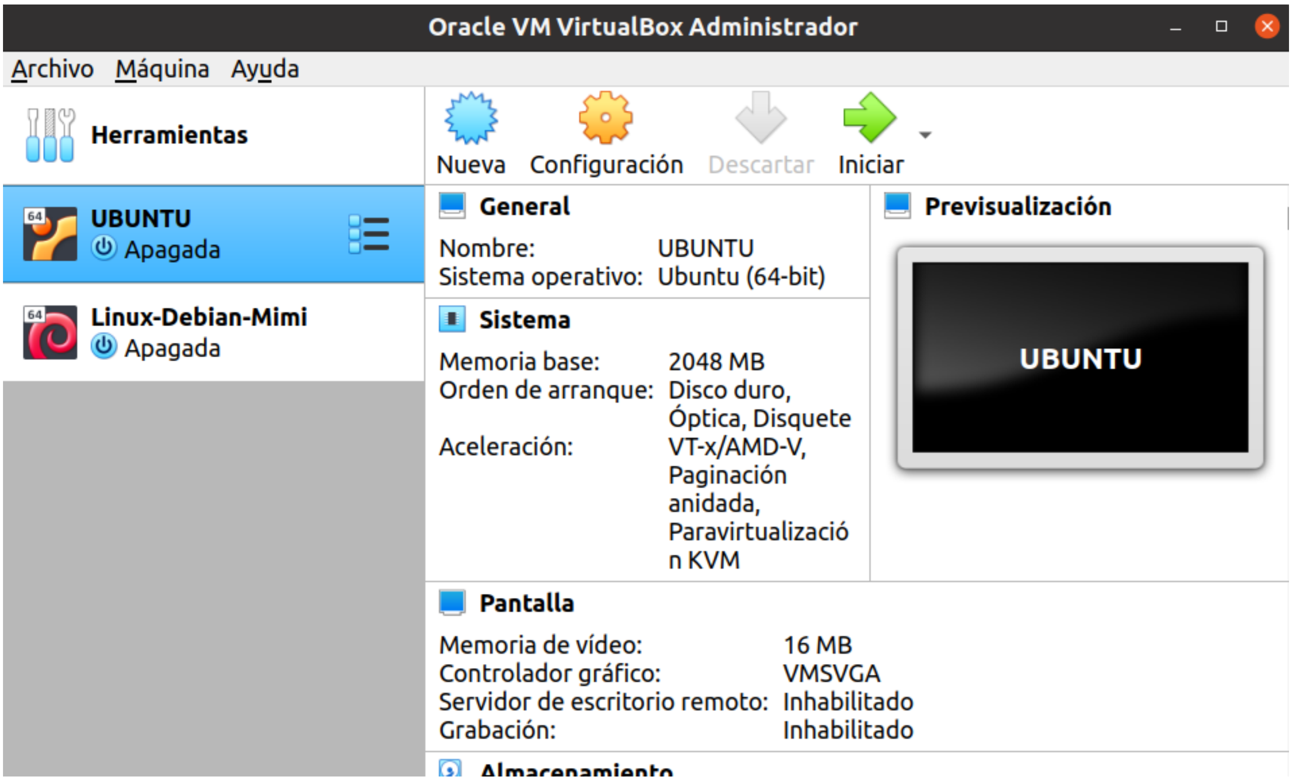


```
mimi@mimi-Lenovo-V330-15IKB:~$ vagrant init ubuntu/focal64
```



```
  A `Vagrantfile` has been placed in this directory. You are now
  ready to `vagrant up` your first virtual environment! Please read
  the comments in the Vagrantfile as well as documentation on
  `vagrantup.com` for more information on using Vagrant.
```

Comprobaciones → VM VirtualBox antes de crear la nueva:



General

Nombre: UBUNTU
Sistema operativo: Ubuntu (64-bit)

Sistema

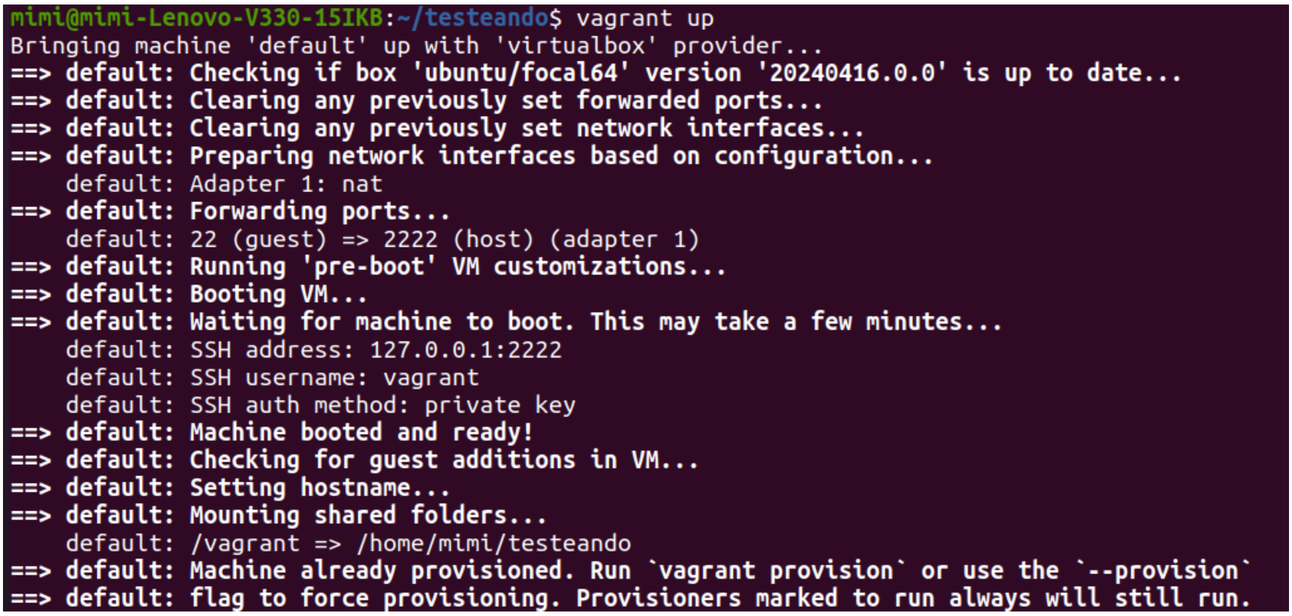
Memoria base: 2048 MB
Orden de arranque: Disco duro, Óptica, Disquete
Aceleración: VT-x/AMD-V, Paginación anidada, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB
Controlador gráfico: VMSVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

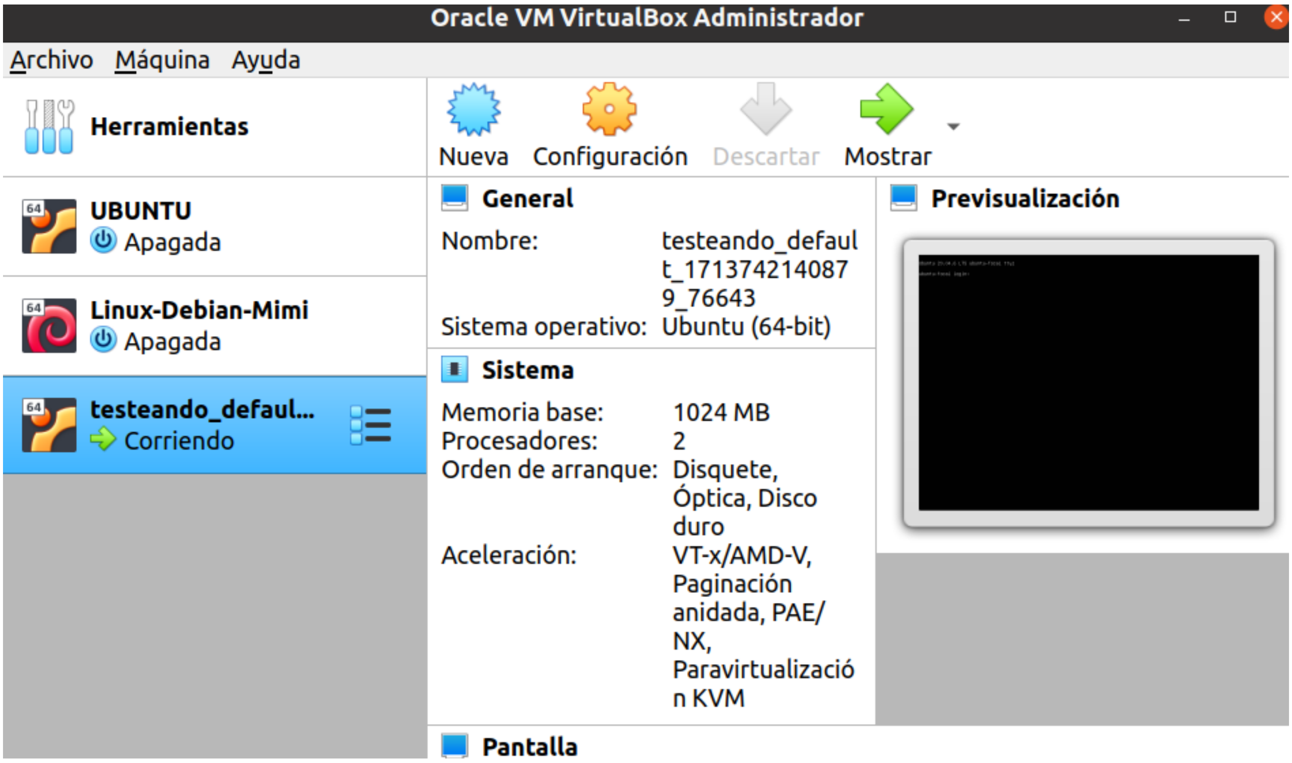
Almacenamiento

Con el comando vagrant up generamos la VM desde el vagranfile antes configurado:



```
mimi@mimi-Lenovo-V330-15IKB:~/testeando$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'ubuntu/focal64' version '20240416.0.0' is up to date...
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Setting hostname...
==> default: Mounting shared folders...
    default: /vagrant => /home/mimi/testeando
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
```

Y confirmamos la creación en la interfaz de VM VirtualBox:



UBUNTU
Apagada

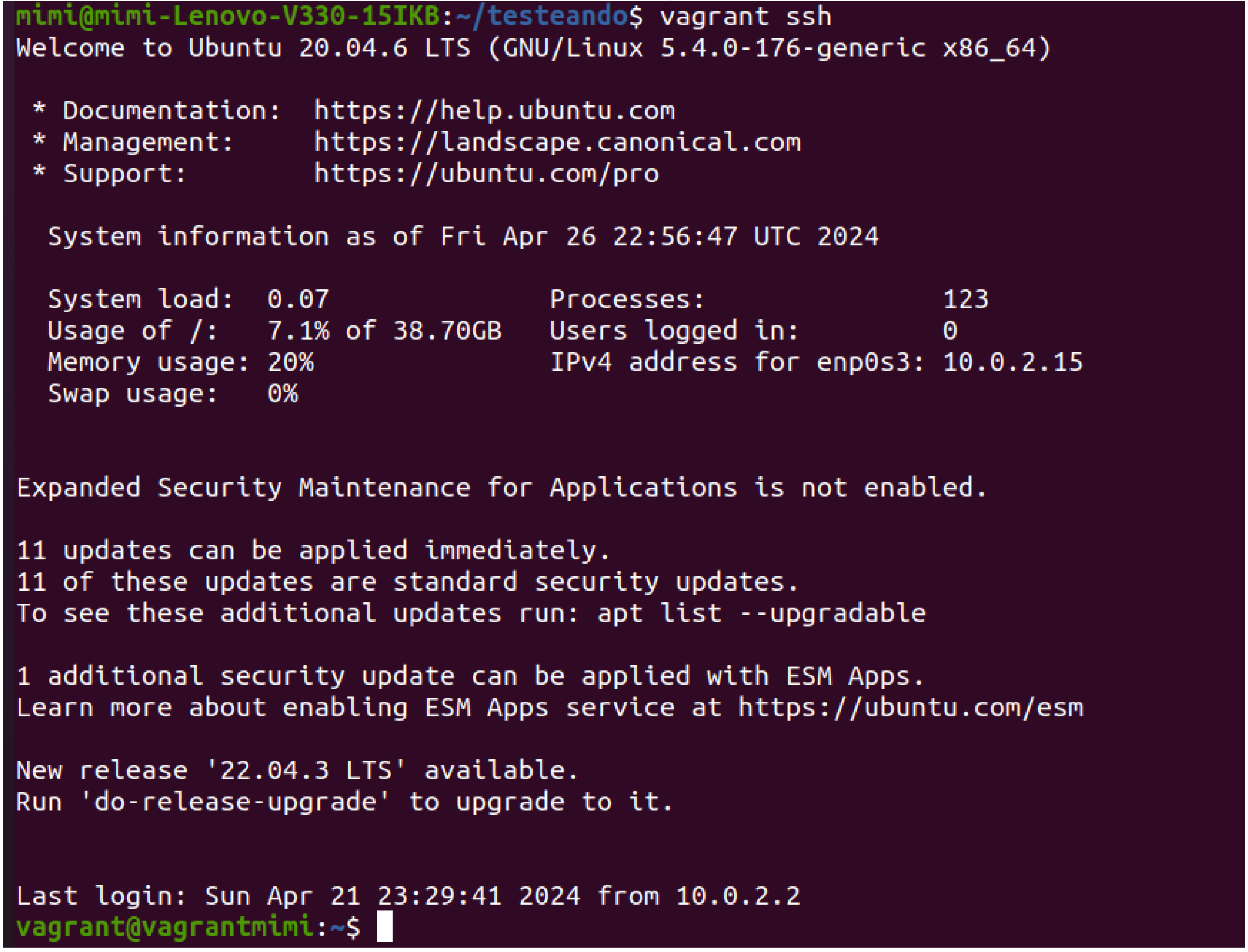
Linux-Debian-Mimi
Apagada

testeando_defaul...
Corriendo

Nombre: testeando_default_1713742140879_76643
Sistema operativo: Ubuntu (64-bit)

Memoria base: 1024 MB
Procesadores: 2
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM

Accedemos a ella mediante SSH con el comando «vagrant ssh»:

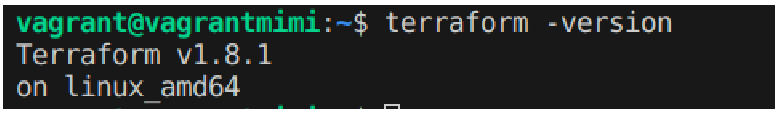


```
mimi@mimi-Lenovo-V330-15IKB:~/testeando$ vagrant ssh
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-176-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Fri Apr 26 22:56:47 UTC 2024

  System load:  0.07              Processes:             123
  Usage of /:   7.1% of 38.70GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sun Apr 21 23:29:41 2024 from 10.0.2.2
vagrant@vagrantmimi:~$
```

| ***Etapa 2: Terraform*** |

Enlace que usé para la instalación de Terraform:
https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

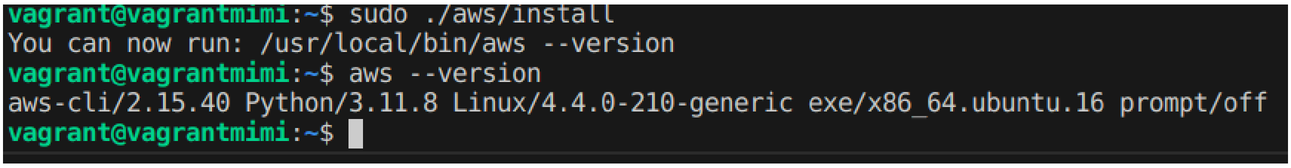
Instalamos Terraform en nuestra VM:



```
vagrant@vagrantmimi:~$ terraform -version
Terraform v1.8.1
on linux_amd64
```

Para no tener las claves de acceso de AWS como texto plano en el archivo de Terraform usaremos la herramienta AWS CLI. Se instala:

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install



```
vagrant@vagrantmimi:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
vagrant@vagrantmimi:~$ aws --version
aws-cli/2.15.40 Python/3.11.8 Linux/4.4.0-210-generic exe/x86_64.ubuntu.16 prompt/off
vagrant@vagrantmimi:~$
```

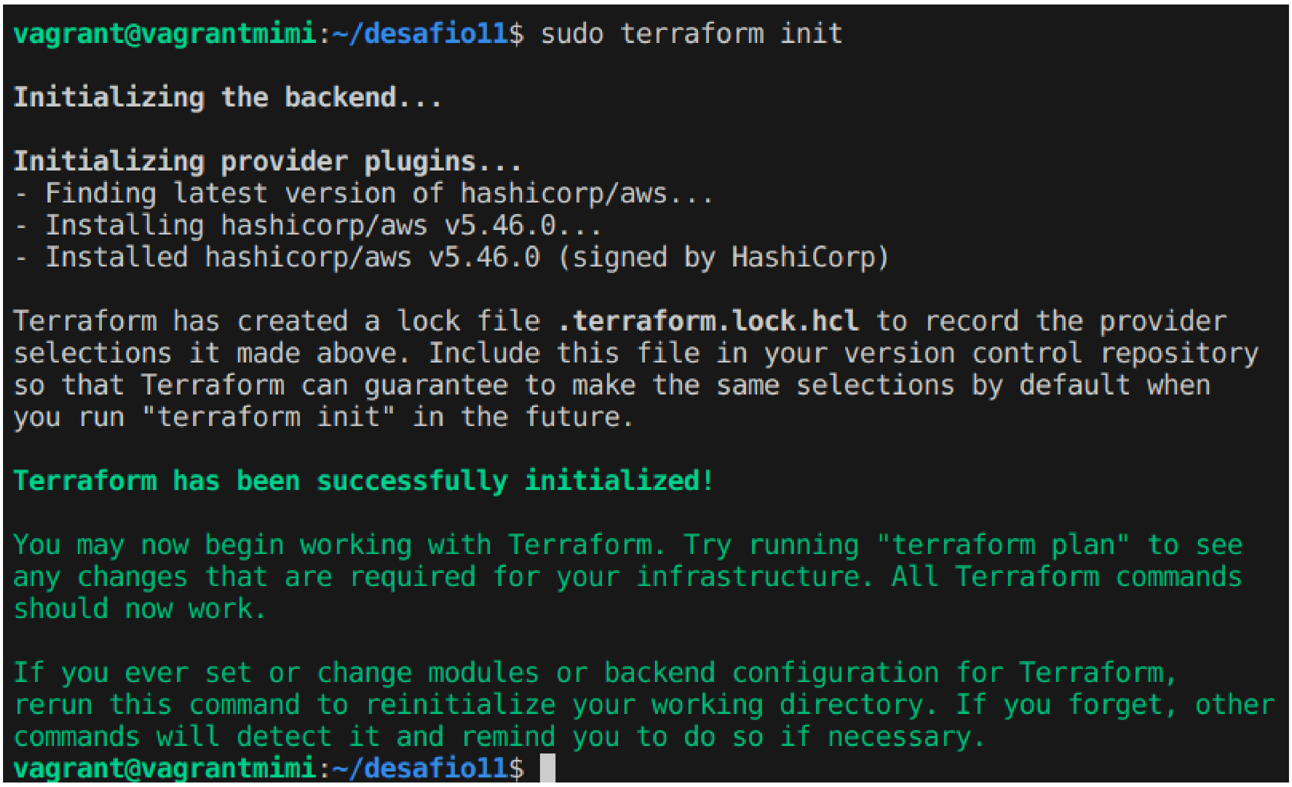
Referencia: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
Para instalar el unzip:
sudo apt update
sudo apt install unzip

Debemos ir al IAM y crear un clave de acceso y utilizaremos el Secret Key – Access Key
En la instancia utilizamos el comando aws configure y accedemos con el Acccess Key y el Secret Access Key y definimos la region us-east-1 y formato (json).

Para el despliegue de Terraform utilizamos VPC y un security group en AWS.

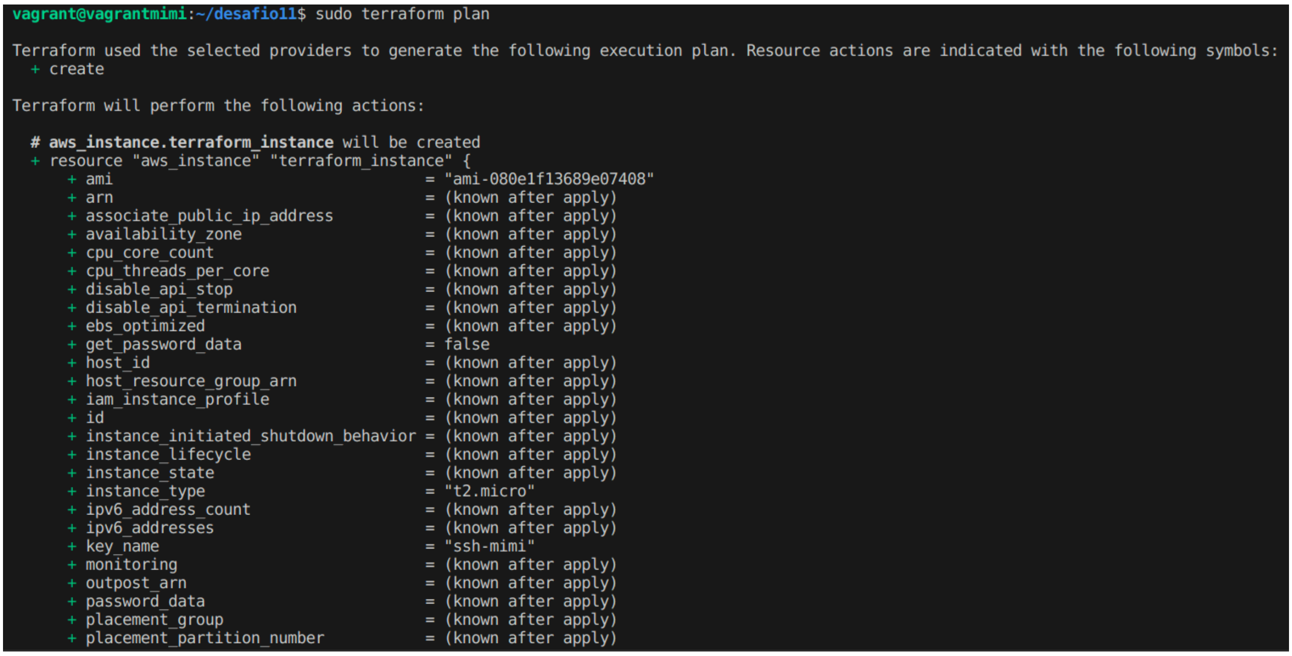
Creamos una carpeta en la ruta /home/vagrant llamada desafio11 y dentro creamos un archivo desafio11.tf en donde se aloja la información de despliegue de la instancia.

Comandos de Terraform:
terraform init → sirve para iniciar el proceso de verificacion de provider en nuestro codigo.

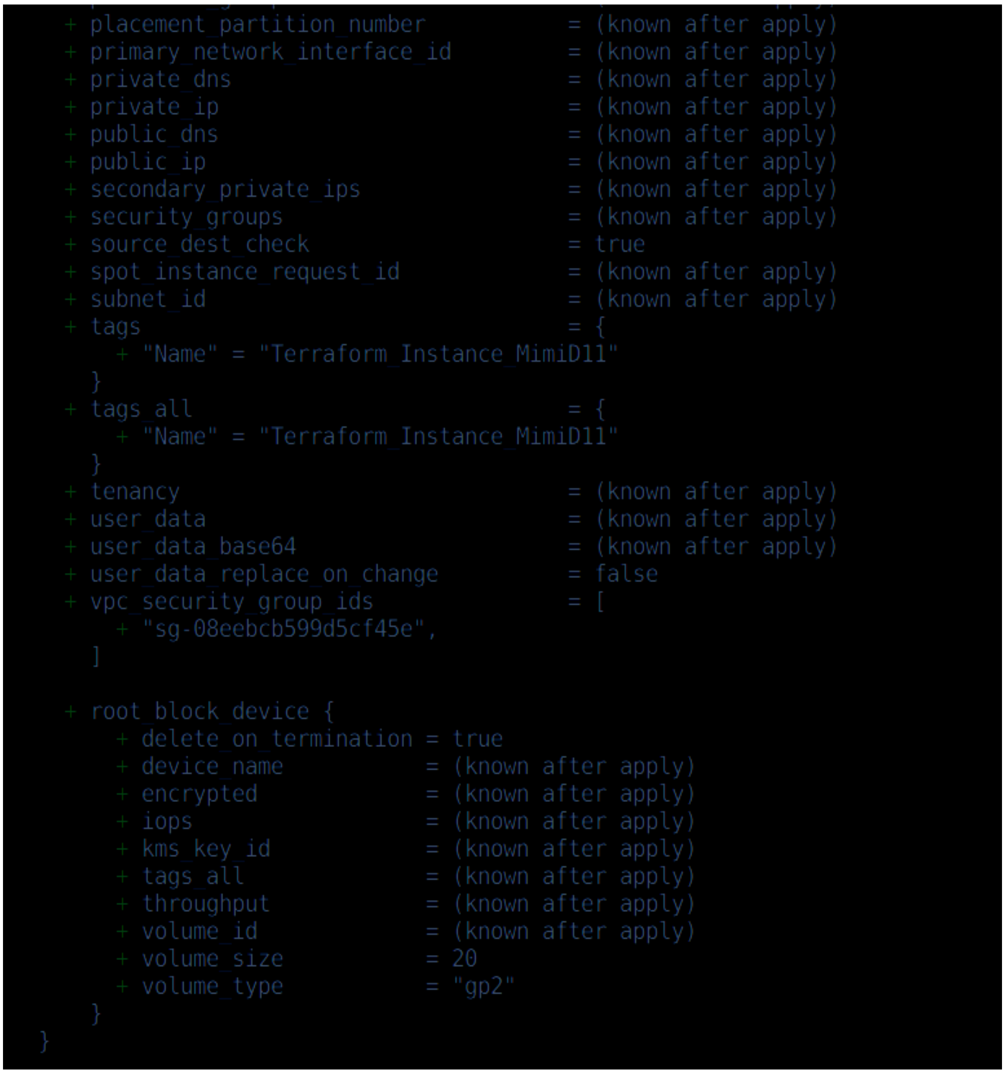


```
vagrant@vagrantmimi:~/desafio11$ sudo terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.46.0...
- Installed hashicorp/aws v5.46.0 (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.
vagrant@vagrantmimi:~/desafio11$
```

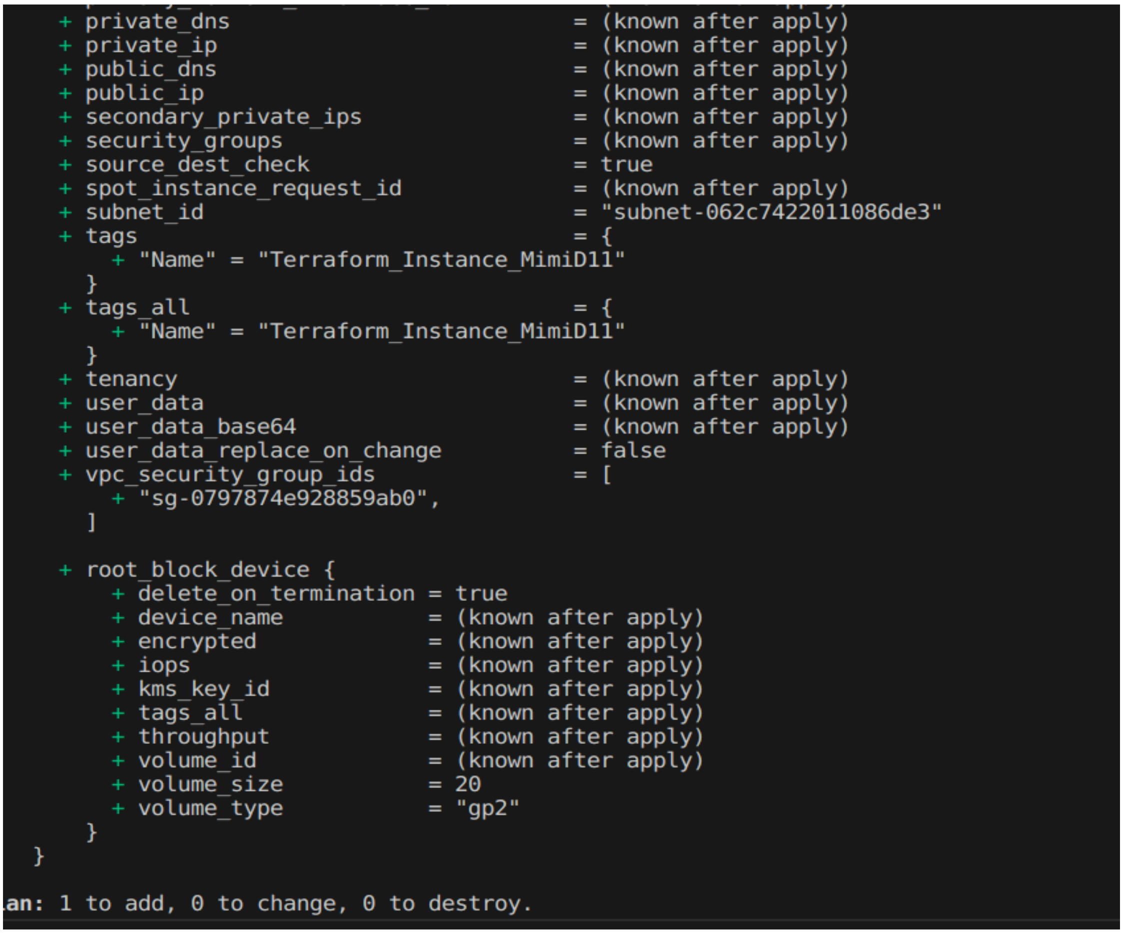
terraform plan → para verificar todo lo que se va a hacer desde nuestro código



```
vagrant@vagrantmimi:~/desafio11$ sudo 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.terraform_instance will be created
  + resource "aws_instance" "terraform_instance" {
      + ami                                  = "ami-080e1f13689e07408"
      + 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)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (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                             = "ssh-mimi"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
```

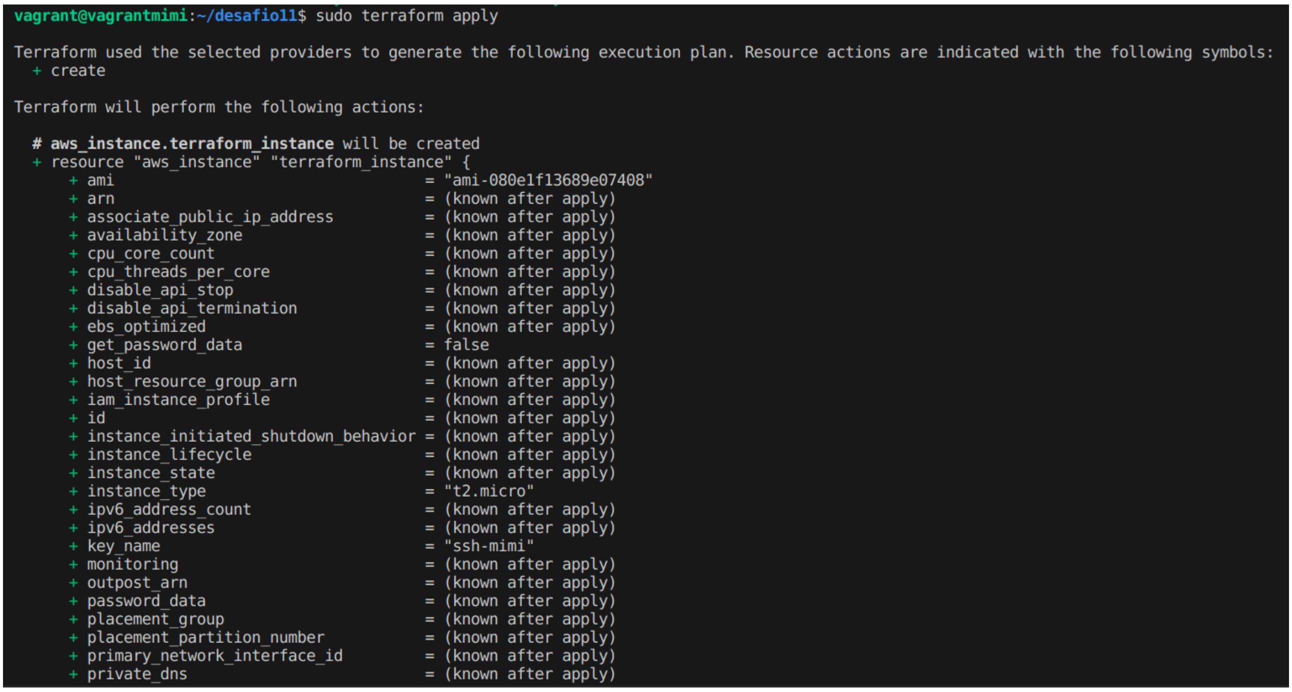


```
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "Terraform_Instance_MimiD11"
        }
      + tags_all                             = {
          + "Name" = "Terraform_Instance_MimiD11"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = [
          + "sg-08eebcb599d5cf45e",
        ]

      + root_block_device {
          + delete_on_termination = true
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags_all              = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = 20
          + volume_type           = "gp2"
        }
    }
```

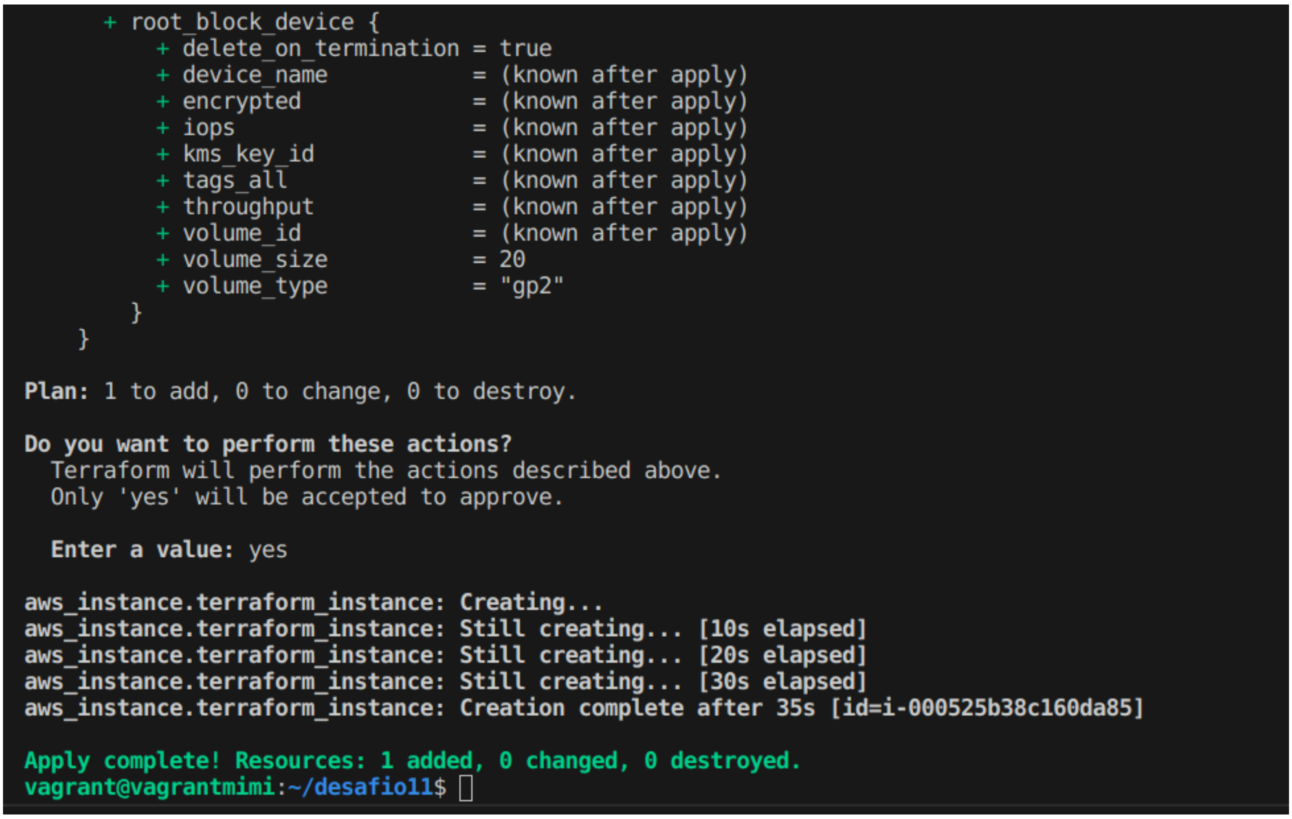


```
    + private_dns                           = (known after apply)
    + private_ip                            = (known after apply)
    + public_dns                            = (known after apply)
    + public_ip                             = (known after apply)
    + secondary_private_ips                 = (known after apply)
    + security_groups                       = (known after apply)
    + source_dest_check                     = true
    + spot_instance_request_id              = (known after apply)
    + subnet_id                             = "subnet-062c7422011086de3"
    + tags                                  = {
        + "Name" = "Terraform_Instance_MimiD11"
      }
    + tags_all                              = {
        + "Name" = "Terraform_Instance_MimiD11"
      }
    + tenancy                               = (known after apply)
    + user_data                             = (known after apply)
    + user_data_base64                      = (known after apply)
    + user_data_replace_on_change           = false
    + vpc_security_group_ids                = [
        + "sg-0797874e928859ab0",
      ]

    + root_block_device {
        + delete_on_termination = true
        + device_name           = (known after apply)
        + encrypted             = (known after apply)
        + iops                  = (known after apply)
        + kms_key_id            = (known after apply)
        + tags_all              = (known after apply)
        + throughput            = (known after apply)
        + volume_id             = (known after apply)
        + volume_size           = 20
        + volume_type           = "gp2"
      }
  }

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

terraform apply → para aplicar o correr nuestro código

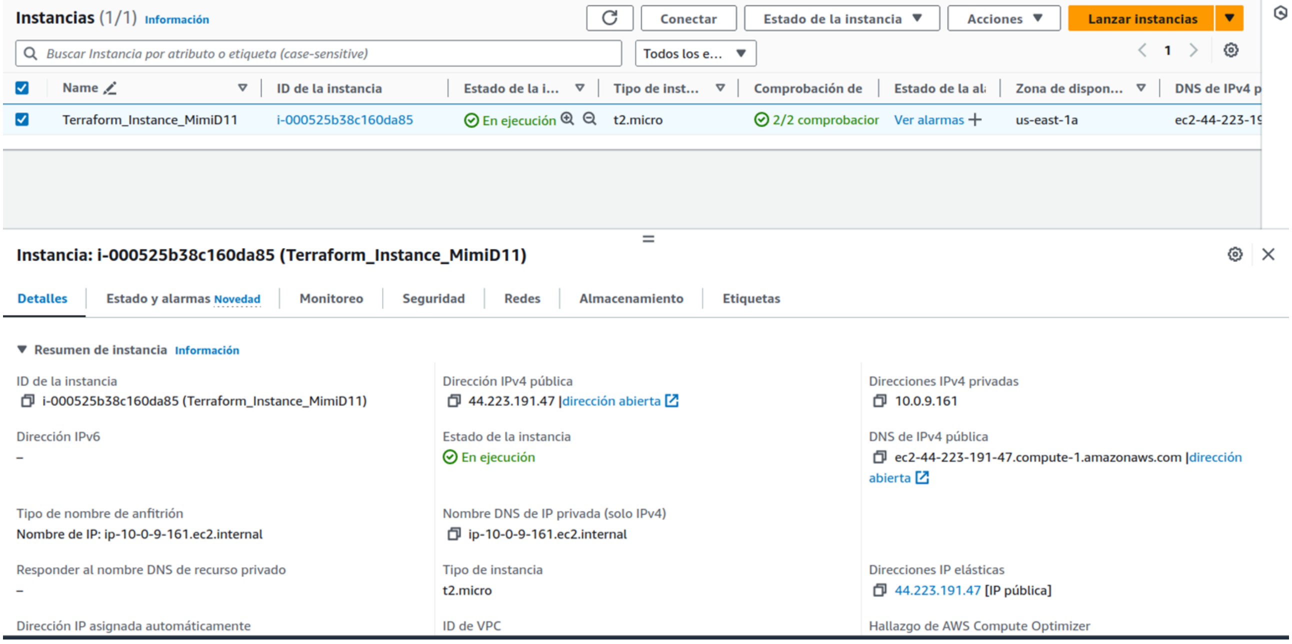


```
vagrant@vagrantmimi:~/desafio11$ sudo terraform apply

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.terraform_instance will be created
  + resource "aws_instance" "terraform_instance" {
      + ami                                  = "ami-080e1f13689e07408"
      + 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)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (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                             = "ssh-mimi"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
```



```
      + root_block_device {
          + delete_on_termination = true
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags_all              = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = 20
          + volume_type           = "gp2"
        }
    }

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.terraform_instance: Creating...
aws_instance.terraform_instance: Still creating... [10s elapsed]
aws_instance.terraform_instance: Still creating... [20s elapsed]
aws_instance.terraform_instance: Still creating... [30s elapsed]
aws_instance.terraform_instance: Creation complete after 35s [id=i-000525b38c160da85]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
vagrant@vagrantmimi:~/desafio11$ 
```

Se comprueba la creación de la instancia:



Instancias (1/1) Información
Conectar
Estado de la instancia ▼
Acciones ▼
Lanzar instancias ▼
Buscar Instancia por atributo o etiqueta (case-sensitive)
Todos los e... ▼
< 1 >

☑	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de	Estado de la al	Zona de dispon...	DNS de IPv4 p
☑	Terraform_Instance_MimiD11	i-000525b38c160da85	✓ En ejecución	t2.micro	✓ 2/2 comprobacior	Ver alarmas +	us-east-1a	ec2-44-223-19

Instancia: i-000525b38c160da85 (Terraform_Instance_MimiD11)

▼ Resumen de instancia Información

ID de la instancia
i-000525b38c160da85 (Terraform_Instance_MimiD11)

Dirección IPv4 pública
44.223.191.47 |dirección abierta

Direcciones IPv4 privadas
10.0.9.161

Dirección IPv6
–

Estado de la instancia
✓ En ejecución

DNS de IPv4 pública
ec2-44-223-191-47.compute-1.amazonaws.com |dirección abierta

Tipo de nombre de anfitrión
Nombre de IP: ip-10-0-9-161.ec2.internal

Nombre DNS de IP privada (solo IPv4)
ip-10-0-9-161.ec2.internal

Responder al nombre DNS de recurso privado
–

Tipo de instancia
t2.micro

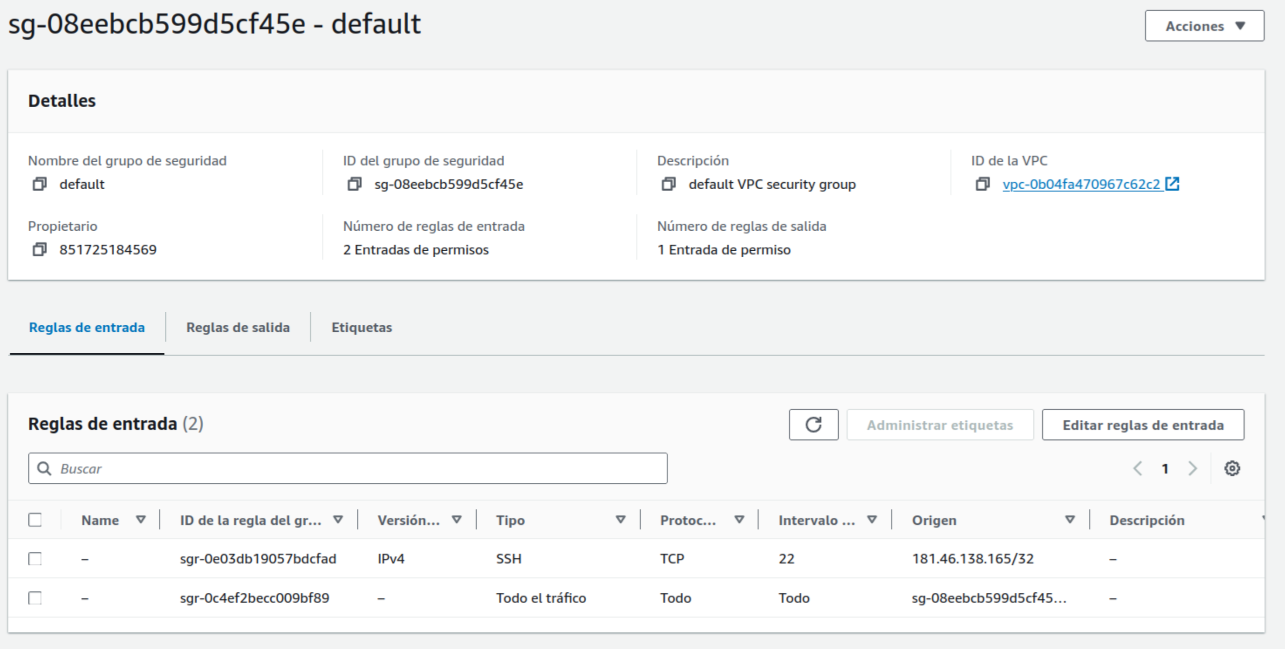
Direcciones IP elásticas
44.223.191.47 [IP pública]

Dirección IP asignada automáticamente

ID de VPC

Hallazgo de AWS Compute Optimizer

Grupo de seguridad:



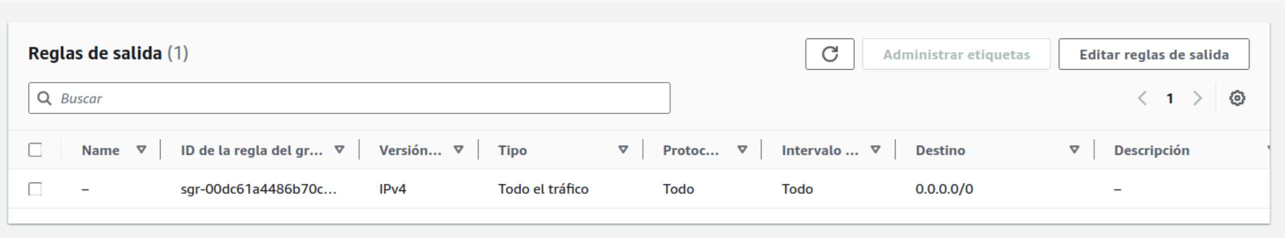
sg-08eebcb599d5cf45e - default

Detalles

Nombre del grupo de seguridad default	ID del grupo de seguridad sg-08eebcb599d5cf45e	Descripción default VPC security group	ID de la VPC vpc-0b04fa470967c62c2
Propietario 851725184569	Número de reglas de entrada 2 Entradas de permisos	Número de reglas de salida 1 Entrada de permiso	

Reglas de entrada (2)

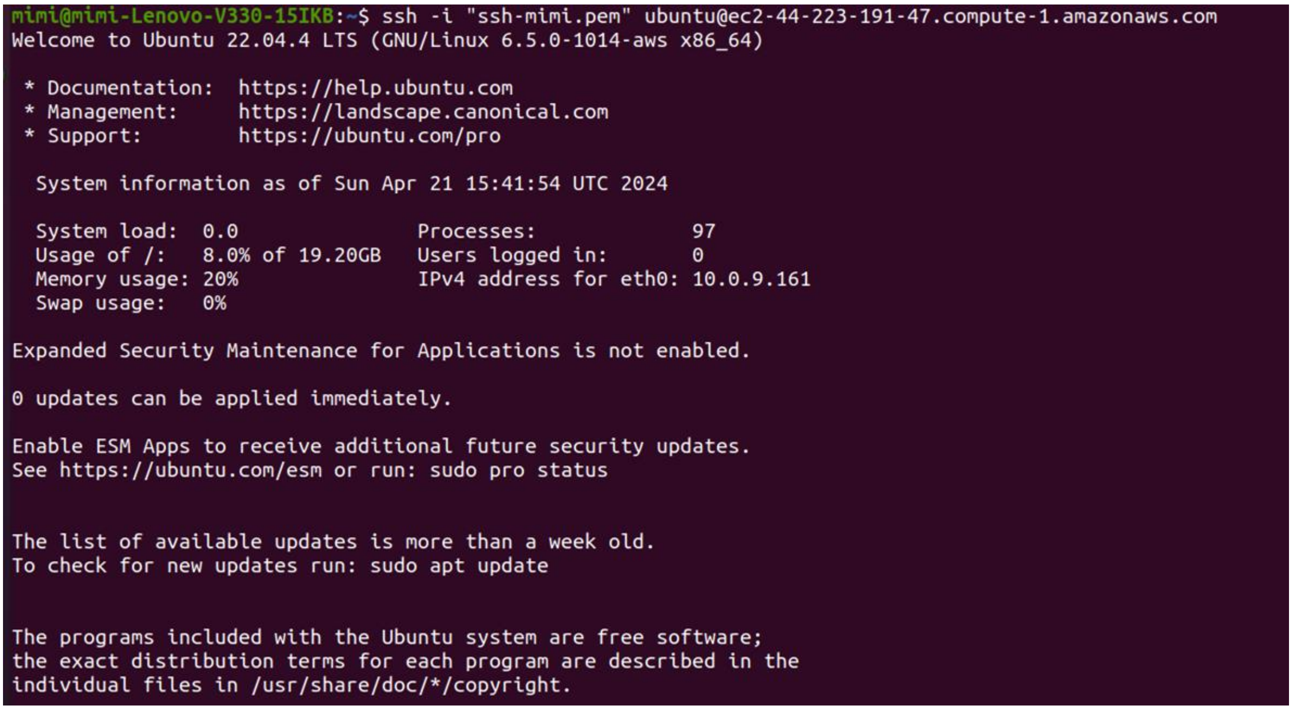
□	Name	ID de la regla del gr...	Versión...	Tipo	Protoc...	Intervalo ...	Origen	Descripción
□	–	sgr-0e03db19057bdcfad	IPv4	SSH	TCP	22	181.46.138.165/32	–
□	–	sgr-0c4ef2becc009bf89	–	Todo el tráfico	Todo	Todo	sg-08eebcb599d5cf45...	–



Reglas de salida (1)

□	Name	ID de la regla del gr...	Versión...	Tipo	Protoc...	Intervalo ...	Destino	Descripción
□	–	sgr-00dc61a4486b70c...	IPv4	Todo el tráfico	Todo	Todo	0.0.0.0/0	–

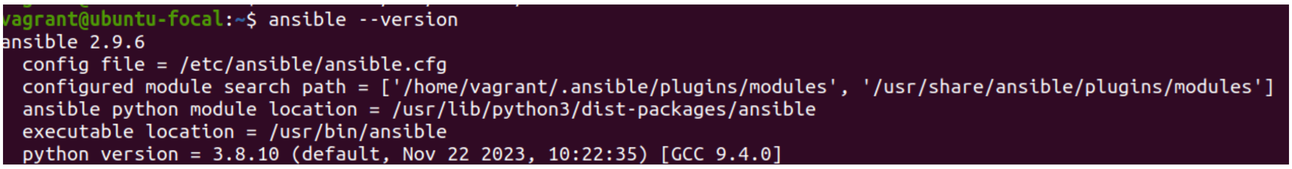
Conexión SSH:



```
mimi@mimi-Lenovo-V330-15IKB:~$ ssh -i "ssh-mimi.pem" ubuntu@ec2-44-223-191-47.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Sun Apr 21 15:41:54 UTC 2024

  System load:  0.0               Processes:             97
  Usage of /:   8.0% of 19.20GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 10.0.9.161
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

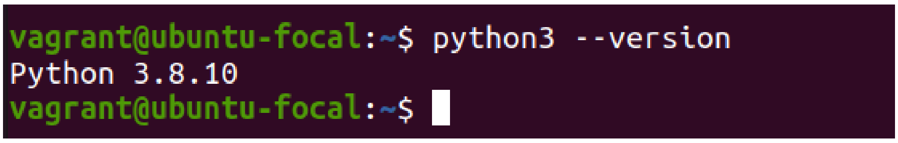
| *Etapa 3: Ansible* |

Instalación de Ansible - Referencia: https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-20-04-es



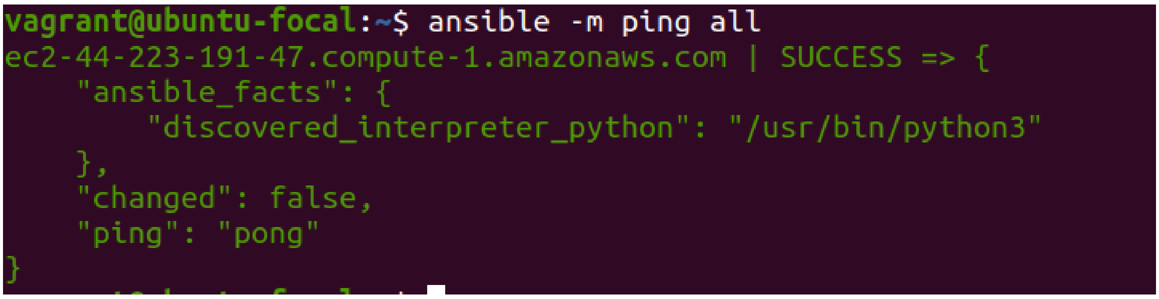
```
vagrant@ubuntu-focal:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vagrant/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 22 2023, 10:22:35) [GCC 9.4.0]
```

Instalamos Python
sudo apt update
sudo apt install python3



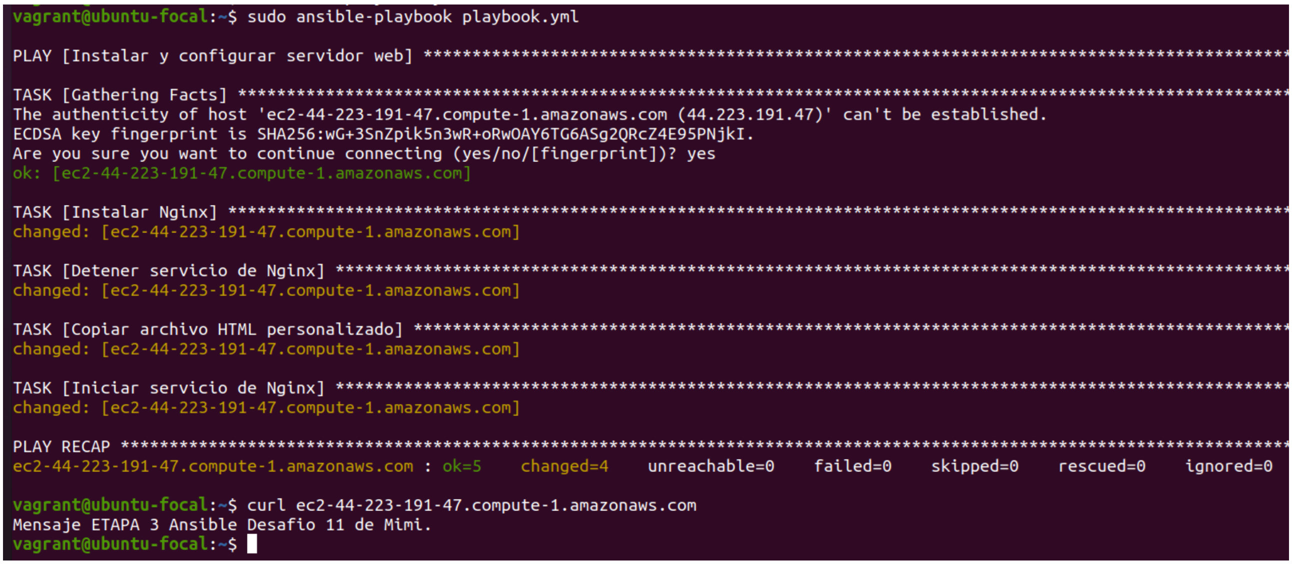
```
vagrant@ubuntu-focal:~$ python3 --version
Python 3.8.10
vagrant@ubuntu-focal:~$
```

Ping a la instancia de aws



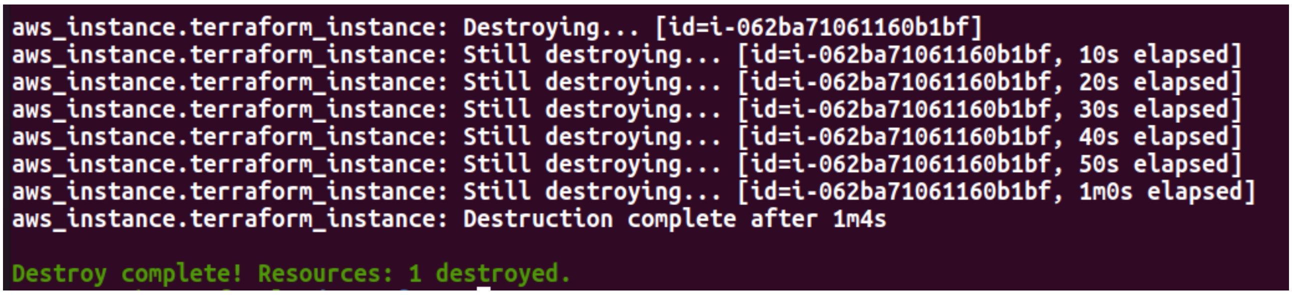
```
vagrant@ubuntu-focal:~$ ansible -m ping all
ec2-44-223-191-47.compute-1.amazonaws.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Creación de un playbook que instale un servidor web, y cambiar el html para que al probar el servidor web nos muestre otro mensaje: *Mensaje ETAPA 3 Ansible Desafio 11 de Mimi.*



```
vagrant@ubuntu-focal:~$ sudo ansible-playbook playbook.yml

PLAY [Instalar y configurar servidor web] *************************************************************************

TASK [Gathering Facts] ********************************************************************************************
The authenticity of host 'ec2-44-223-191-47.compute-1.amazonaws.com (44.223.191.47)' can't be established.
ECDSA key fingerprint is SHA256:wG+3SnZpik5n3wR+oRwOAY6TG6ASg2QRcZ4E95PNjkI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
ok: [ec2-44-223-191-47.compute-1.amazonaws.com]

TASK [Instalar Nginx] *********************************************************************************************
changed: [ec2-44-223-191-47.compute-1.amazonaws.com]

TASK [Detener servicio de Nginx] **********************************************************************************
changed: [ec2-44-223-191-47.compute-1.amazonaws.com]

TASK [Copiar archivo HTML personalizado] **************************************************************************
changed: [ec2-44-223-191-47.compute-1.amazonaws.com]

TASK [Iniciar servicio de Nginx] **********************************************************************************
changed: [ec2-44-223-191-47.compute-1.amazonaws.com]

PLAY RECAP ********************************************************************************************************
ec2-44-223-191-47.compute-1.amazonaws.com : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vagrant@ubuntu-focal:~$ curl ec2-44-223-191-47.compute-1.amazonaws.com
Mensaje ETAPA 3 Ansible Desafio 11 de Mimi.
vagrant@ubuntu-focal:~$
```

Finalmente utilizamos ***terraform desatroy*** → para destruir toda la infraestructura creada:



```
aws_instance.terraform_instance: Destroying... [id=i-062ba71061160b1bf]
aws_instance.terraform_instance: Still destroying... [id=i-062ba71061160b1bf, 10s elapsed]
aws_instance.terraform_instance: Still destroying... [id=i-062ba71061160b1bf, 20s elapsed]
aws_instance.terraform_instance: Still destroying... [id=i-062ba71061160b1bf, 30s elapsed]
aws_instance.terraform_instance: Still destroying... [id=i-062ba71061160b1bf, 40s elapsed]
aws_instance.terraform_instance: Still destroying... [id=i-062ba71061160b1bf, 50s elapsed]
aws_instance.terraform_instance: Still destroying... [id=i-062ba71061160b1bf, 1m0s elapsed]
aws_instance.terraform_instance: Destruction complete after 1m4s

Destroy complete! Resources: 1 destroyed.
```