UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Laboratorio 3 - ANTLR

Edwin Andrés Ortega Kou – 22305

Esteban Zambrano Garoz – 22119

Juan Pablo Solís Albizurez – 22102

Bianca Renata Calderon Caravantes – 22272

Daniel Eduardo Dubon Ortiz – 22233

Construcción de Compiladores

Guatemala, julio 2025

Configuración de Terraform

```
PS C:\Users\Andy Ortega\Progras\Compis\Terraform_Compis2\terraform> 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:

  # digitalocean_droplet.web will be created
  + resource "digitalocean_droplet" "web" {
      + backups              = false
      + created_at           = (known after apply)
      + disk                 = (known after apply)
      + graceful_shutdown    = false
      + id                   = (known after apply)
      + image                = "ubuntu-24-10-x64"
      + ipv4_address         = (known after apply)
      + ipv4_address_private = (known after apply)
      + ipv6                 = false
      + ipv6_address         = (known after apply)
      + locked               = (known after apply)
      + memory               = (known after apply)
      + monitoring           = false
      + name                 = "example-droplet"
      + price_hourly         = (known after apply)
      + price_monthly        = (known after apply)
      + private_networking   = (known after apply)
      + region               = "nyc1"
      + resize_disk          = true
      + size                 = "s-1vcpu-512mb-10gb"
      + status               = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + vpc_uuid             = (known after apply)
    }

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

Changes to Outputs:
  + droplet_ip = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
```

```
      + size                 = "s-1vcpu-512mb-10gb"
      + status               = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + vpc_uuid             = (known after apply)
    }

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

Changes to Outputs:
  + droplet_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

digitalocean_droplet.web: Creating...
digitalocean_droplet.web: Still creating... [00m10s elapsed]
digitalocean_droplet.web: Still creating... [00m20s elapsed]
digitalocean_droplet.web: Creation complete after 23s [id=510681560]

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

Outputs:

droplet_ip = "134.209.221.192"
PS C:\Users\Andy Ortega\Progras\Compis\Terraform_Compis2\terraform> ping 134.209.221.192

Pinging 134.209.221.192 with 32 bytes of data:
Reply from 134.209.221.192: bytes=32 time=57ms TTL=46
Reply from 134.209.221.192: bytes=32 time=57ms TTL=46
Reply from 134.209.221.192: bytes=32 time=56ms TTL=46
Reply from 134.209.221.192: bytes=32 time=56ms TTL=46

Ping statistics for 134.209.221.192:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 56ms, Maximum = 57ms, Average = 56ms
```

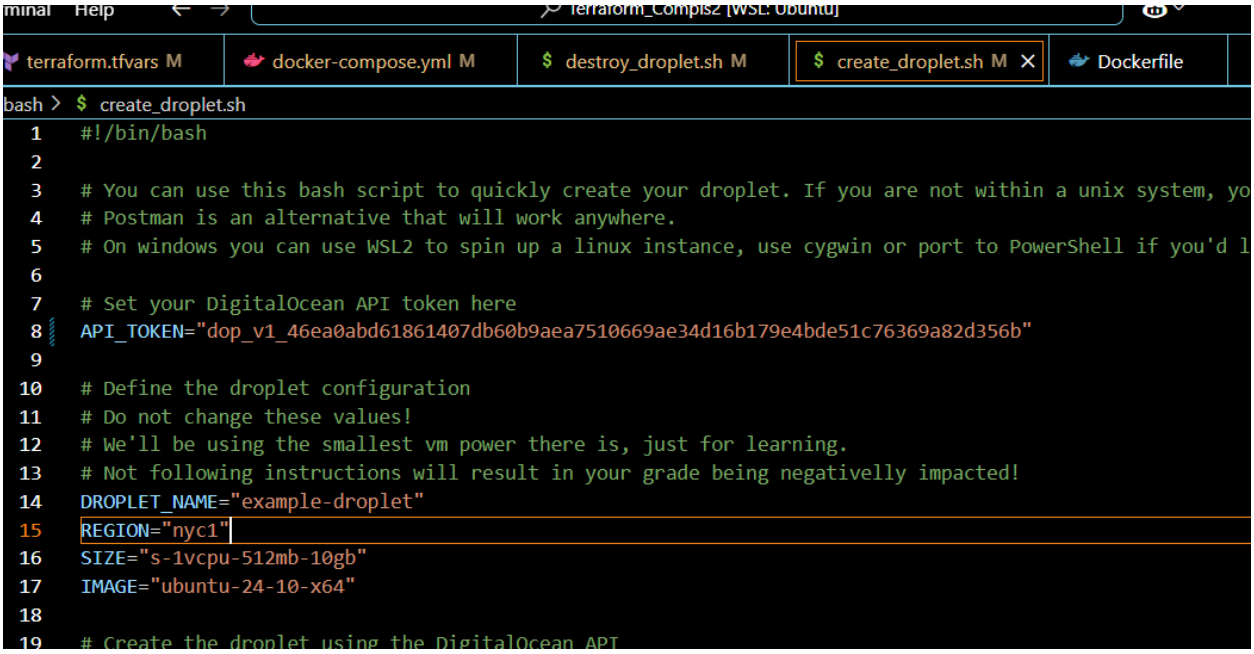
```
PS C:\Users\Andy Ortega\Progras\Compis\Terraform_Compis2\terraform> terraform destroy
digitalocean_droplet.web: Refreshing state... [id=510681560]

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:

  # digitalocean_droplet.web will be destroyed
  - resource "digitalocean_droplet" "web" {
      - backups              = false -> null
      - created_at           = "2025-07-29T23:15:21Z" -> null
      - disk                 = 10 -> null
      - graceful_shutdown    = false -> null
      - id                   = "510681560" -> null
      - image                = "ubuntu-24-10-x64" -> null
      - ipv4_address         = "134.209.221.192" -> null
      - ipv4_address_private = "10.116.0.20" -> null
      - ipv6                 = false -> null
      - locked               = false -> null
      - memory               = 512 -> null
      - monitoring           = false -> null
      - name                 = "example-droplet" -> null
      - price_hourly         = 0.00595 -> null
      - price_monthly        = 4 -> null
      - private_networking   = true -> null
      - region               = "nyc1" -> null
      - resize_disk          = true -> null
      - size                 = "s-1vcpu-512mb-10gb" -> null
      - status               = "active" -> null
      - tags                 = [] -> null
      - urn                  = "do:droplet:510681560" -> null
      - vcpus                = 1 -> null
      - volume_ids           = [] -> null
      - vpc_uuid             = "2ddc24ca-204a-46fe-8b73-06ce664c2cb5" -> null
        # (1 unchanged attribute hidden)
    }

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

```
      - status               = "active" -> null
      - tags                 = [] -> null
      - urn                  = "do:droplet:510681560" -> null
      - vcpus                = 1 -> null
      - volume_ids           = [] -> null
      - vpc_uuid             = "2ddc24ca-204a-46fe-8b73-06ce664c2cb5" -> null
        # (1 unchanged attribute hidden)
    }

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

Changes to Outputs:
  - droplet_ip = "134.209.221.192" -> 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

digitalocean_droplet.web: Destroying... [id=510681560]
digitalocean_droplet.web: Still destroying... [id=510681560, 00m10s elapsed]
digitalocean_droplet.web: Still destroying... [id=510681560, 00m20s elapsed]
digitalocean_droplet.web: Destruction complete after 24s

Destroy complete! Resources: 1 destroyed.
```

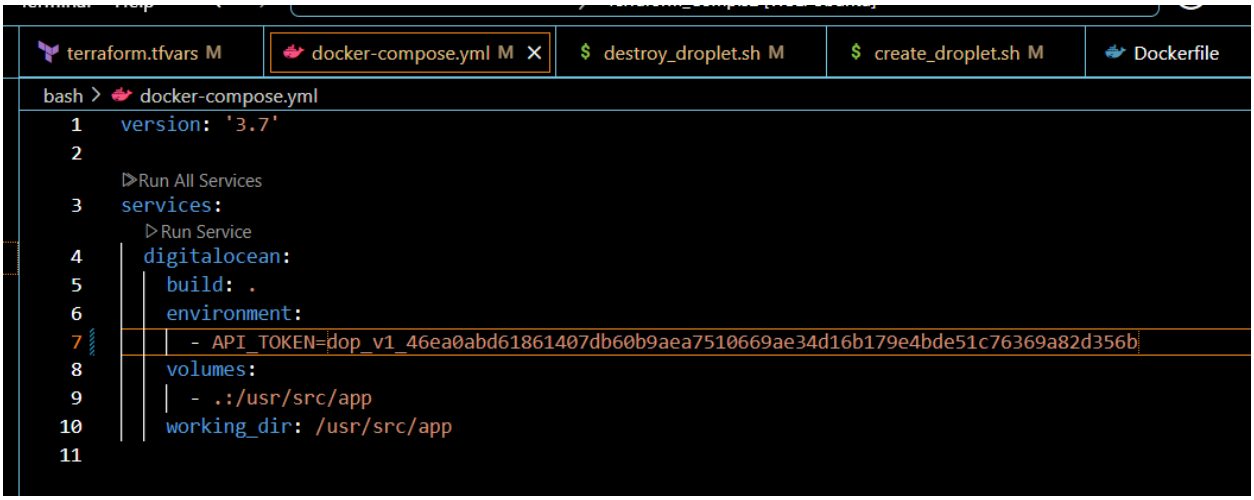
Configuración de Bash y Docker

```
laptopez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ docker-compose build
[+] Building 53.2s (11/11) FINISHED                                                                                 docker:default
 => [digitalocean internal] load .dockerignore                                                                                0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [digitalocean internal] load build definition from Dockerfile                                                             0.0s
 => => transferring dockerfile: 574B                                                                                          0.0s
 => [digitalocean internal] load metadata for docker.io/library/ubuntu:20.04                                                  0.7s
 => CACHED [digitalocean 1/6] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c42             0.0s
 => [digitalocean internal] load build context                                                                                0.0s
 => => transferring context: 77B                                                                                              0.0s
 => [digitalocean 2/6] RUN apt-get update && apt-get install -y     curl     jq     && rm -rf /var/lib/apt/lists           51.2s
 => [digitalocean 3/6] COPY create_droplet.sh /usr/local/bin/create_droplet.sh                                                0.1s
 => [digitalocean 4/6] COPY destroy_droplet.sh /usr/local/bin/destroy_droplet.sh                                              0.1s
 => [digitalocean 5/6] RUN chmod +x /usr/local/bin/create_droplet.sh                                                          0.4s
 => [digitalocean 6/6] RUN chmod +x /usr/local/bin/destroy_droplet.sh                                                         0.4s
 => [digitalocean] exporting to image                                                                                         0.2s
 => => exporting layers                                                                                                       0.2s
 => => writing image sha256:e19e3540f2e39d88858b55d047159d610386a1a9ce803eb226ffe8f09780f480                                  0.0s
 => => naming to docker.io/library/bash-digitalocean                                                                          0.0s
laptopez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$
```

```
laptopez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ docker-compose run digitalocean /usr/local/bin/create_dr
oplet.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3975  100  3878  100    97   3481     87  0:00:01  0:00:01 --:--:--  3568
Droplet created with ID: 510720269
laptopez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ docker-compose run digitalocean /usr/local/bin/destroy_d
roplet.sh
Droplet with ID 510720269 has been destroyed
laptopez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$
```

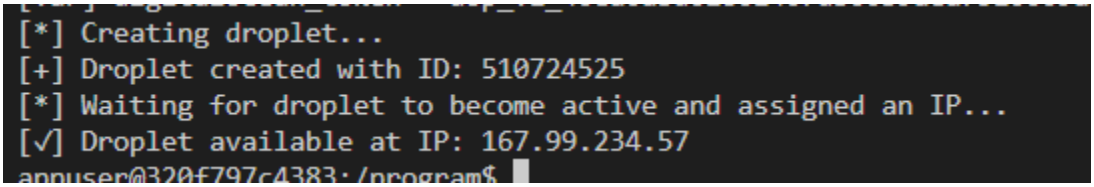



```bash
#!/bin/bash

# You MUST destroy all droplets.

# Set your DigitalOcean API token here
API_TOKEN="dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b"

# Read the droplet ID from the file
DROPLET_ID=$(cat droplet_id.txt)

# Destroy the droplet using the DigitalOcean API
curl -k -X DELETE "https://api.digitalocean.com/v2/droplets/$DROPLET_ID" \
    -H "Authorization: Bearer $API_TOKEN"

echo "Droplet with ID $DROPLET_ID has been destroyed"
```

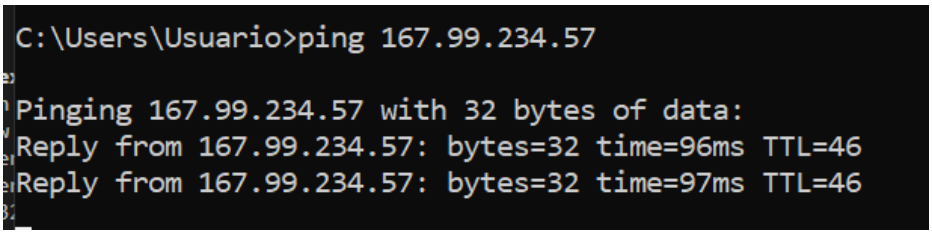


Uso de ANTLR para parsear Terraform y Administrar Droplets de Digital Ocean

Compilacion de Archivo Terraform usando el driver



Ping a la IP



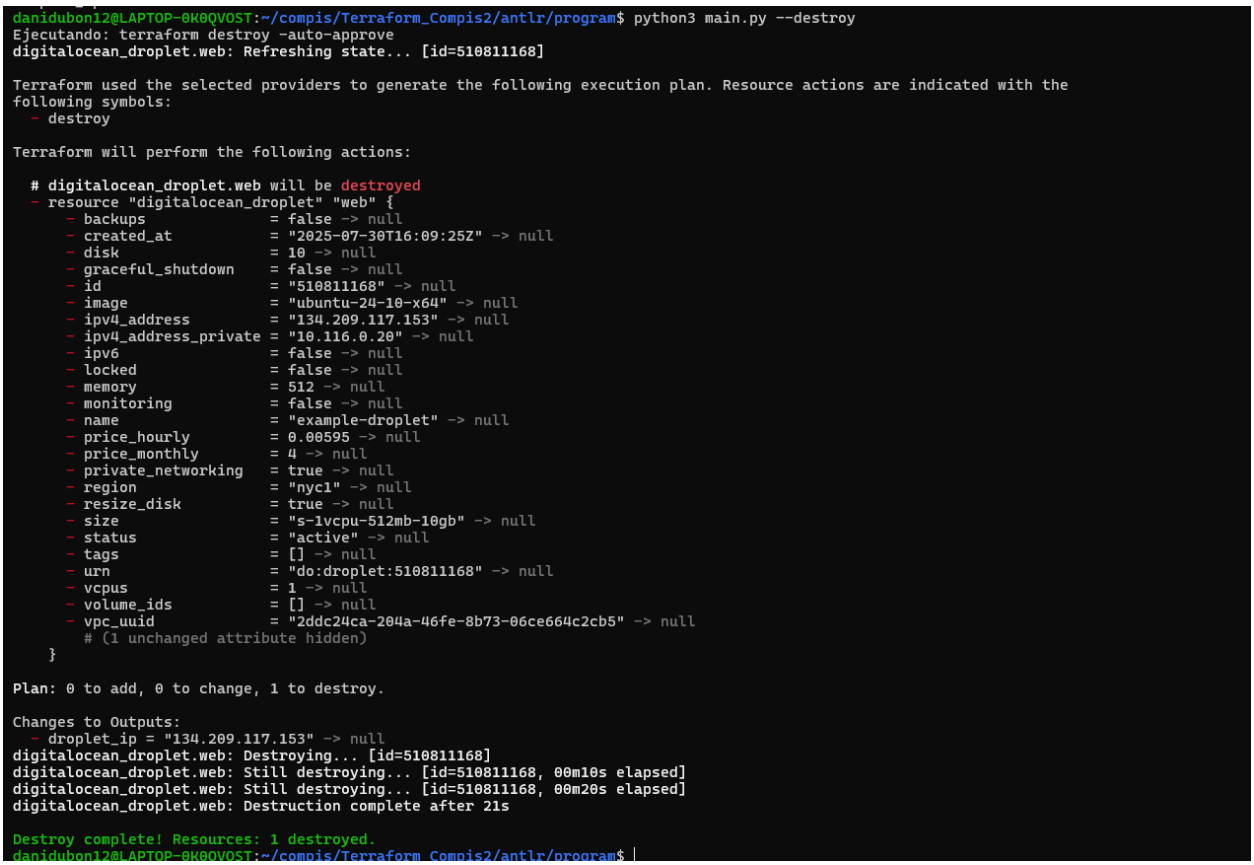
Destroy Droplet

```
laptopez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ export API_TOKEN=dop_v1_46ea0abd61861407db60b9aea7510669
ae34d16b179e4bde51c76369a82d356b

# 2. Elimina el droplet por ID
curl -X DELETE \
     -H "Authorization: Bearer $API_TOKEN" \
     "https://api.digitalocean.com/v2/droplets/510724525"
```

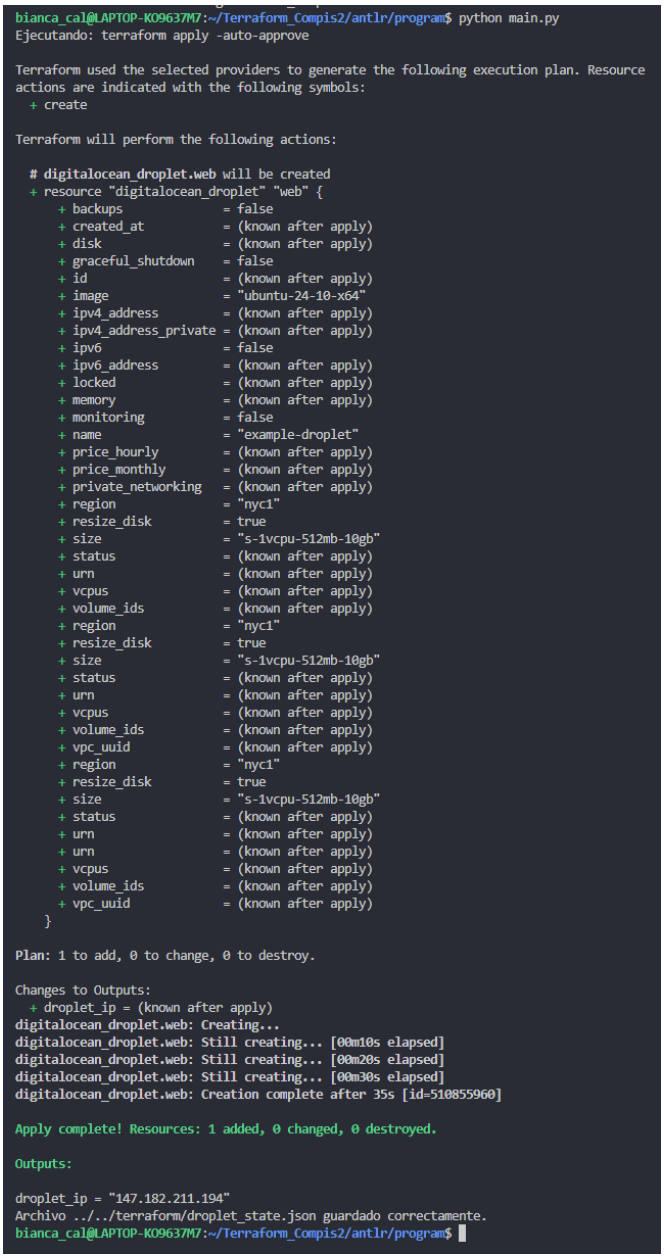
Ejecución de Terraform Apply desde script Python

```
danidubon12@LAPTOP-0K0QVOST:~/compis/Terraform_Compis2/antlr/program$ python3 main.py
Ejecutando: terraform apply -auto-approve

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:

  # digitalocean_droplet.web will be created
  + resource "digitalocean_droplet" "web" {
      + backups              = false
      + created_at           = (known after apply)
      + disk                 = (known after apply)
      + graceful_shutdown    = false
      + id                   = (known after apply)
      + image                = "ubuntu-24-10-x64"
      + ipv4_address         = (known after apply)
      + ipv4_address_private = (known after apply)
      + ipv6                 = false
      + ipv6_address         = (known after apply)
      + locked               = (known after apply)
      + memory               = (known after apply)
      + monitoring           = false
      + name                 = "example-droplet"
      + price_hourly         = (known after apply)
      + price_monthly        = (known after apply)
      + private_networking   = (known after apply)
      + region               = "nyc1"
      + resize_disk          = true
      + size                 = "s-1vcpu-512mb-10gb"
      + status               = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + vpc_uuid             = (known after apply)
    }

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

Changes to Outputs:
  + droplet_ip = (known after apply)
digitalocean_droplet.web: Creating...
digitalocean_droplet.web: Still creating... [00m10s elapsed]
digitalocean_droplet.web: Still creating... [00m20s elapsed]
digitalocean_droplet.web: Still creating... [00m30s elapsed]
digitalocean_droplet.web: Creation complete after 34s [id=510811168]

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

Outputs:

droplet_ip = "134.209.117.153"
```

Ejecución de Terraform Destroy desde script Python



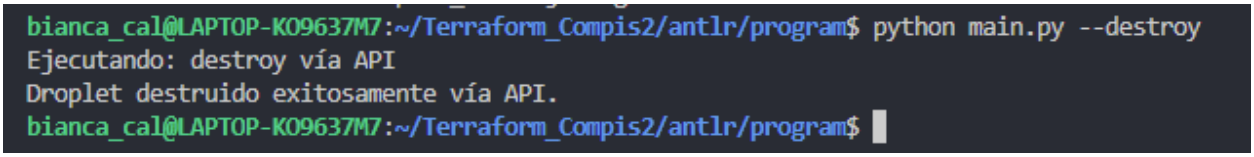
Creación Droplet:



```
bianca_cal@LAPTOP-KO9637M7:~/Terraform_Compis2/antlr/program$ python main.py
Ejecutando: terraform apply -auto-approve

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:

  # digitalocean_droplet.web will be created
  + resource "digitalocean_droplet" "web" {
      + backups              = false
      + created_at           = (known after apply)
      + disk                 = (known after apply)
      + graceful_shutdown    = false
      + id                   = (known after apply)
      + image                = "ubuntu-24-10-x64"
      + ipv4_address         = (known after apply)
      + ipv4_address_private = (known after apply)
      + ipv6                 = false
      + ipv6_address         = (known after apply)
      + locked               = (known after apply)
      + memory               = (known after apply)
      + monitoring           = false
      + name                 = "example-droplet"
      + price_hourly         = (known after apply)
      + price_monthly        = (known after apply)
      + private_networking   = (known after apply)
      + region               = "nyc1"
      + resize_disk          = true
      + size                 = "s-1vcpu-512mb-10gb"
      + status               = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + region               = "nyc1"
      + resize_disk          = true
      + size                 = "s-1vcpu-512mb-10gb"
      + status               = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + vpc_uuid             = (known after apply)
      + region               = "nyc1"
      + resize_disk          = true
      + size                 = "s-1vcpu-512mb-10gb"
      + status               = (known after apply)
      + urn                  = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + vpc_uuid             = (known after apply)
    }

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

Changes to Outputs:
  + droplet_ip = (known after apply)
digitalocean_droplet.web: Creating...
digitalocean_droplet.web: Still creating... [00m10s elapsed]
digitalocean_droplet.web: Still creating... [00m20s elapsed]
digitalocean_droplet.web: Still creating... [00m30s elapsed]
digitalocean_droplet.web: Creation complete after 35s [id=510855960]

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

Outputs:

droplet_ip = "147.182.211.194"
Archivo ../../terraform/droplet_state.json guardado correctamente.
bianca_cal@LAPTOP-KO9637M7:~/Terraform_Compis2/antlr/program$
```

Se crea archivo .json:

Destruir Droplet con script:



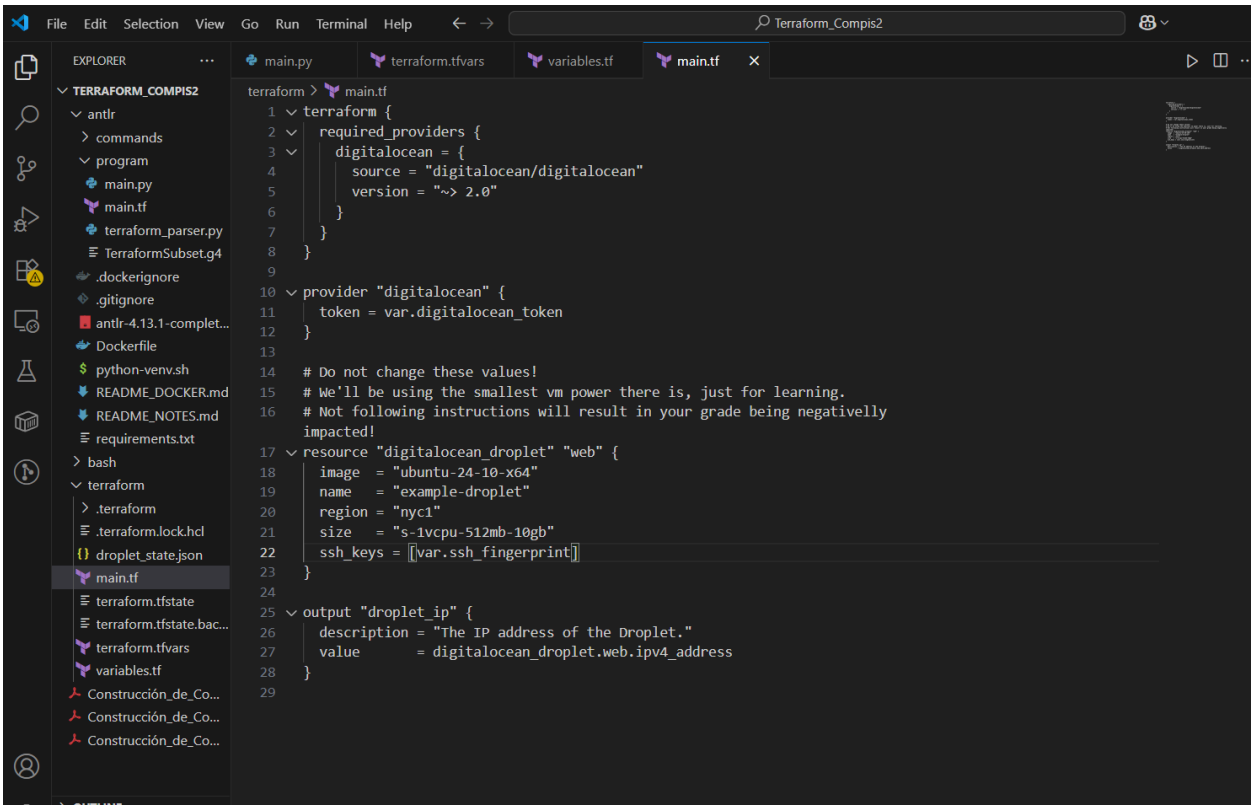
Conexión exitosa por SSH al Droplet creado desde Terraform

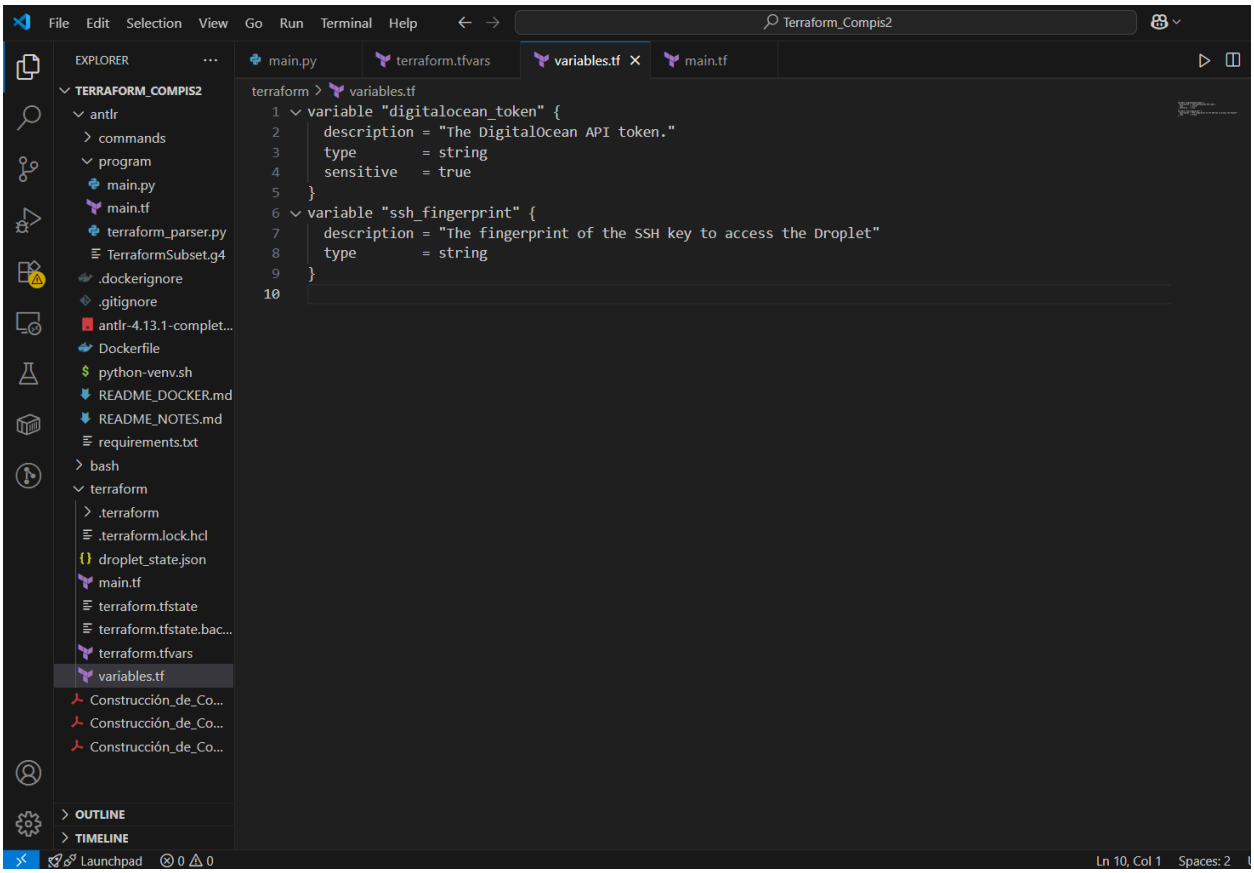
Durante la implementación de los puntos extra relacionados con el uso de llaves SSH en Terraform, inicialmente se presentó un error al intentar crear el Droplet:

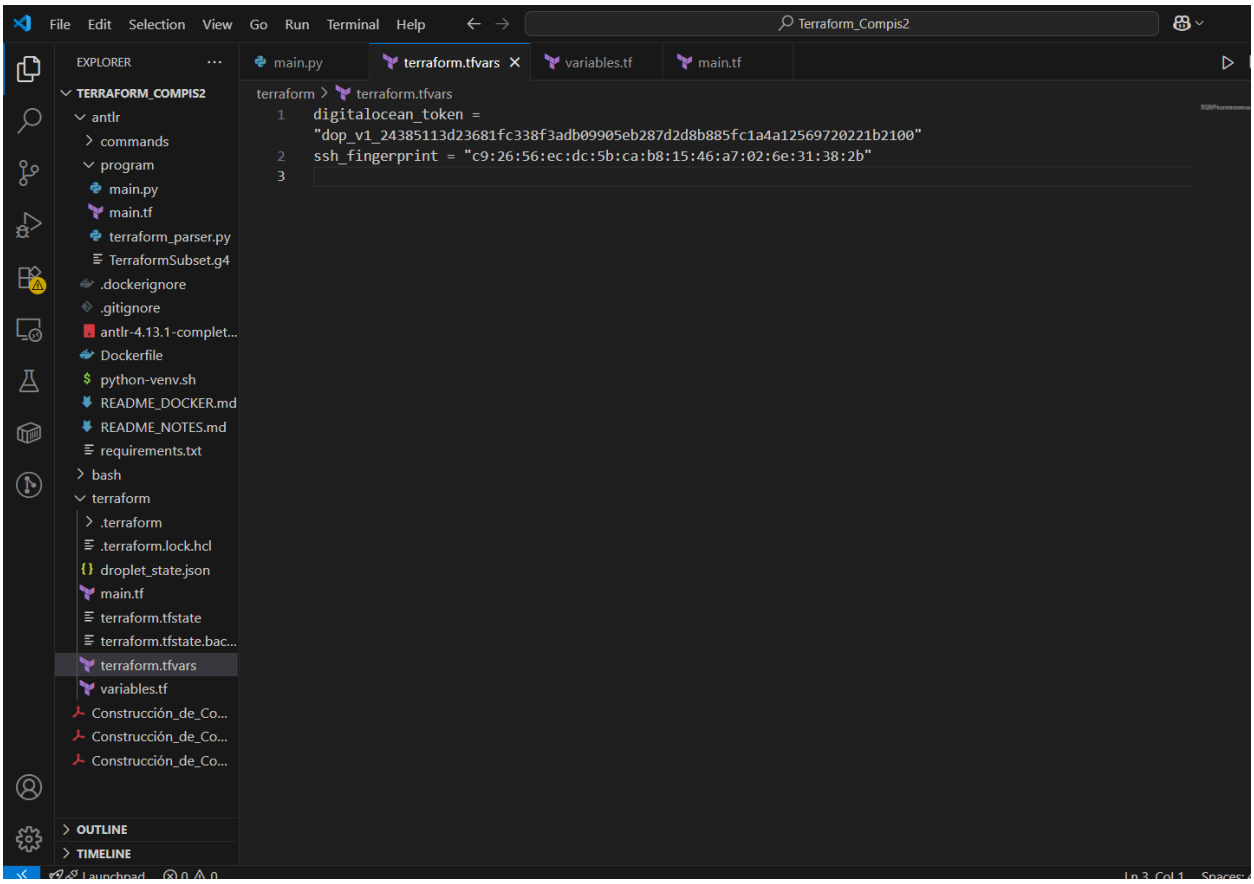
"You are missing the required permission ssh_key:read"

Este error indicaba que el token de acceso de la API de DigitalOcean no tenía permisos suficientes para leer las llaves SSH registradas en la cuenta.

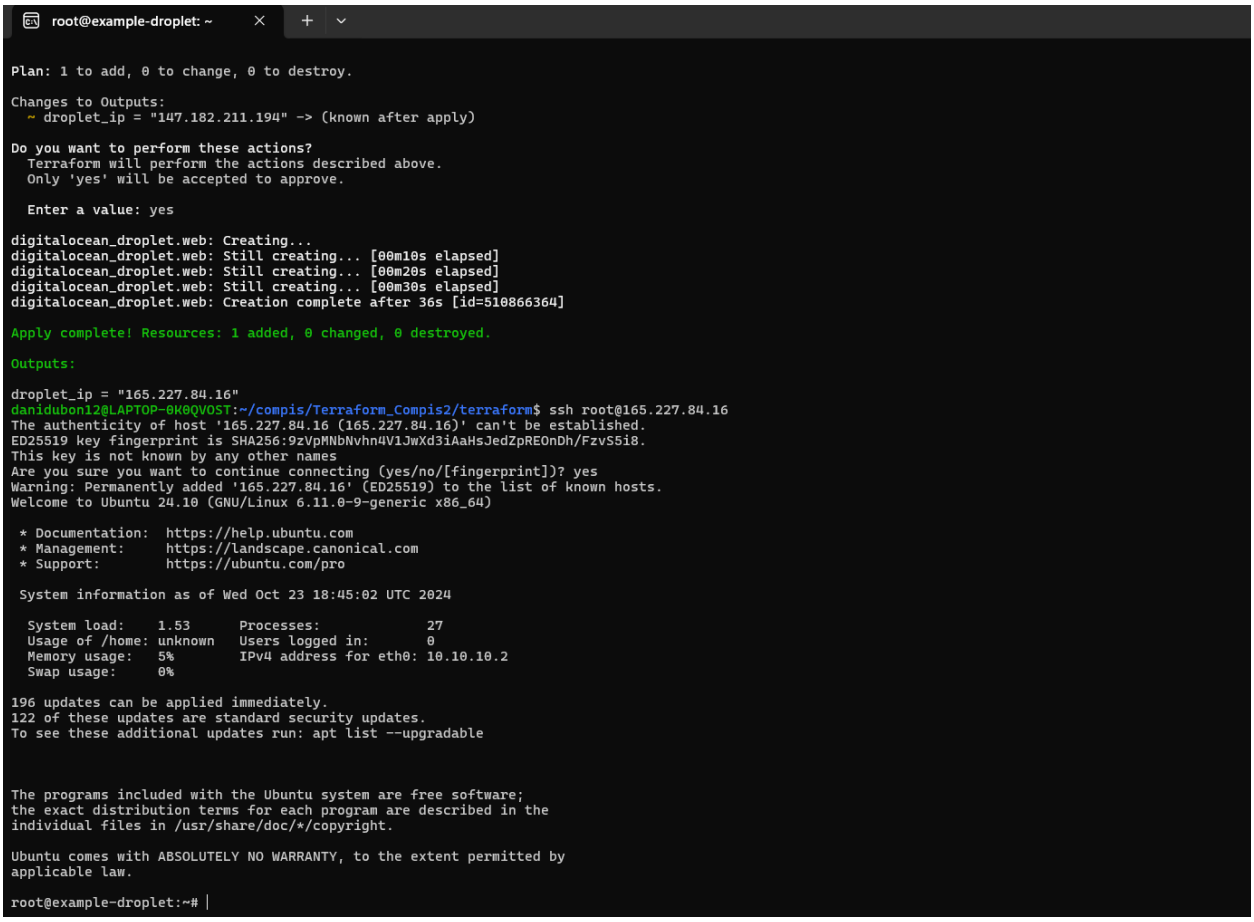
Para solucionarlo, fue necesario generar una nueva API key desde el panel de DigitalOcean, asegurándome de habilitar los permisos "Read and Write". Esto otorgó el acceso necesario para que Terraform pudiera vincular la llave SSH al crear el recurso, completando así exitosamente la conexión remota por ssh.


File Edit Selection View Go Run Terminal Help
Terraform_Compis2
EXPLORER
TERRAFORM_COMPIS2
antlr
commands
program
main.py
main.tf
terraform_parser.py
TerraformSubset.g4
.dockerignore
.gitignore
antlr-4.13.1-complet...
Dockerfile
python-venv.sh
README_DOCKER.md
README_NOTES.md
requirements.txt
bash
terraform
.terraform
.terraform.lock.hcl
droplet_state.json
main.tf
terraform.tfstate
terraform.tfstate.bac...
terraform.tfvars
variables.tf
Construcción_de_Co...
Construcción_de_Co...
Construcción_de_Co...
main.py
terraform.tfvars
variables.tf
main.tf
terraform > main.tf
terraform {
required_providers {
digitalocean = {
source = "digitalocean/digitalocean"
version = "~> 2.0"
}
}
}
provider "digitalocean" {
token = var.digitalocean_token
}
# Do not change these values!
# We'll be using the smallest vm power there is, just for learning.
# Not following instructions will result in your grade being negatively
impacted!
resource "digitalocean_droplet" "web" {
image = "ubuntu-24-10-x64"
name = "example-droplet"
region = "nyc1"
size = "s-1vcpu-512mb-10gb"
ssh_keys = [var.ssh_fingerprint]
}
output "droplet_ip" {
description = "The IP address of the Droplet."
value = digitalocean_droplet.web.ipv4_address
}


File Edit Selection View Go Run Terminal Help
Terraform_Compis2
EXPLORER
TERRAFORM_COMPIS2
antlr
commands
program
main.py
main.tf
terraform_parser.py
TerraformSubset.g4
.dockerignore
.gitignore
antlr-4.13.1-complet...
Dockerfile
python-venv.sh
README_DOCKER.md
README_NOTES.md
requirements.txt
bash
terraform
.terraform
.terraform.lock.hcl
droplet_state.json
main.tf
terraform.tfstate
terraform.tfstate.bac...
terraform.tfvars
variables.tf
Construcción_de_Co...
Construcción_de_Co...
Construcción_de_Co...
OUTLINE
TIMELINE
main.py
terraform.tfvars
variables.tf
main.tf
terraform > variables.tf
variable "digitalocean_token" {
description = "The DigitalOcean API token."
type = string
sensitive = true
}
variable "ssh_fingerprint" {
description = "The fingerprint of the SSH key to access the Droplet"
type = string
}
Launchpad
Ln 10, Col 1
Spaces: 2


File Edit Selection View Go Run Terminal Help
Terraform_Compis2
EXPLORER
TERRAFORM_COMPIS2
antlr
commands
program
main.py
main.tf
terraform_parser.py
TerraformSubset.g4
.dockerignore
.gitignore
antlr-4.13.1-complet...
Dockerfile
python-venv.sh
README_DOCKER.md
README_NOTES.md
requirements.txt
bash
terraform
.terraform
.terraform.lock.hcl
droplet_state.json
main.tf
terraform.tfstate
terraform.tfstate.bac...
terraform.tfvars
variables.tf
Construcción_de_Co...
Construcción_de_Co...
Construcción_de_Co...
OUTLINE
TIMELINE
main.py
terraform.tfvars
variables.tf
main.tf
terraform > terraform.tfvars
1 digitalocean_token =
"dop_v1_24385113d23681fc338f3adb09905eb287d2d8b885fc1a4a12569720221b2100"
2 ssh_fingerprint = "c9:26:56:ec:dc:5b:ca:b8:15:46:a7:02:6e:31:38:2b"
3
Launchpad
Ln 3, Col 1

```
danidubon12@LAPTOP-0K0QVOST:~/compis/Terraform_Compis2/terraform$ 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:

  # digitalocean_droplet.web will be created
  + resource "digitalocean_droplet" "web" {
      + backups              = false
      + created_at           = (known after apply)
      + disk                 = (known after apply)
      + graceful_shutdown    = false
      + id                   = (known after apply)
      + image                = "ubuntu-24-10-x64"
      + ipv4_address         = (known after apply)
      + ipv4_address_private = (known after apply)
      + ipv6                 = false
      + ipv6_address         = (known after apply)
      + locked               = (known after apply)
      + memory               = (known after apply)
      + monitoring           = false
      + name                 = "example-droplet"
      + price_hourly         = (known after apply)
      + price_monthly        = (known after apply)
      + private_networking   = (known after apply)
      + region               = "nyc1"
      + resize_disk          = true
      + size                 = "s-1vcpu-512mb-10gb"
      + ssh_keys             = [
          + "c9:26:56:ec:dc:5b:ca:b8:15:46:a7:02:6e:31:38:2b",
        ]
      + status               = (known after apply)
      + urn                  = (known after apply)
      + vcpus                = (known after apply)
      + volume_ids           = (known after apply)
      + vpc_uuid             = (known after apply)
    }

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

Changes to Outputs:
  ~ droplet_ip = "147.182.211.194" -> (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

digitalocean_droplet.web: Creating...
digitalocean_droplet.web: Still creating... [00m10s elapsed]
digitalocean_droplet.web: Still creating... [00m20s elapsed]
digitalocean_droplet.web: Still creating... [00m30s elapsed]
```



Enlace a video explicatorio

https://youtu.be/ujgbcpZeA38