

# 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)
  + vcpu               = (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)
+ vcpu               = (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

```
laptoppez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ docker-compose build
[+] Building 53.2s (11/11) FINISHED
=> [digitalocean internal] load .dockerignore                                docker:default 0.0s
=> => transferring context: 28                                              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
laptoppez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$
```

```
laptoppez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ docker-compose run digitalocean /usr/local/bin/create_droplet.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
laptoppez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$ docker-compose run digitalocean /usr/local/bin/destroy_droplet.sh
Droplet with ID 510720269 has been destroyed
laptoppez@LaptopEZ:~/4to/segundo/compis2/Terraform_Compis2/bash$
```

```
terminal Help terraform_compsiz [WSL: Ubuntu]
terraform.tfvars M docker-compose.yml M $ destroy_droplet.sh M $ create_droplet.sh M X Dockerfile

bash > $ create_droplet.sh
1  #!/bin/bash
2
3  # You can use this bash script to quickly create your droplet. If you are not within a unix system, yo
4  # Postman is an alternative that will work anywhere.
5  # On windows you can use WSL2 to spin up a linux instance, use cygwin or port to PowerShell if you'd l
6
7  # Set your DigitalOcean API token here
8  API_TOKEN="dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b"
9
10 # Define the droplet configuration
11 # Do not change these values!
12 # We'll be using the smallest vm power there is, just for learning.
13 # Not following instructions will result in your grade being negatively impacted!
14 DROPLET_NAME="example-droplet"
15 REGION="nyc1"
16 SIZE="s-1vcpu-512mb-10gb"
17 IMAGE="ubuntu-24-10-x64"
18
19 # Create the droplet using the DigitalOcean API

1  #!/bin/bash
2
3  # You MUST destroy all droplets.
4
5  # Set your DigitalOcean API token here
6  API_TOKEN="dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b"
7
8  # Read the droplet ID from the file
9  DROPLET_ID=$(cat droplet_id.txt)
10
11 # Destroy the droplet using the DigitalOcean API
12 curl -k -X DELETE "https://api.digitalocean.com/v2/droplets/${DROPLET_ID}" \
13     -H "Authorization: Bearer $API_TOKEN"
14
15 echo "Droplet with ID ${DROPLET_ID} has been destroyed"
16
```

```
terminal Help terraform_compsiz [WSL: Ubuntu]
terraform.tfvars M docker-compose.yml M X $ destroy_droplet.sh M $ create_droplet.sh M Dockerfile

bash > docker-compose.yml
1  version: '3.7'
2
3  >Run All Services
4  services:
5    >Run Service
6    digitalocean:
7      build: .
8      environment:
9        - API_TOKEN=dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b
10     volumes:
11       - ./usr/src/app
12     working_dir: /usr/src/app
```

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

### Compilacion de Archivo Terraform usando el driver

```
[*] Creating droplet...  
[+] Droplet created with ID: 510724525  
[*] Waiting for droplet to become active and assigned an IP...  
[✓] Droplet available at IP: 167.99.234.57  
annuser@320f797c4383: /program$
```

### Ping a la IP

```
C:\Users\Usuario>ping 167.99.234.57  
  
Pinging 167.99.234.57 with 32 bytes of data:  
Reply from 167.99.234.57: bytes=32 time=96ms TTL=46  
Reply from 167.99.234.57: bytes=32 time=97ms TTL=46
```

## Destroy Droplet

```
laptop@LaptopE7:~/4to/segundo/compis2/Terraform_Compis2/bash$ export API_TOKEN=dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b

# 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

```
danidubon12@LAPTOP-0K0QVOST:~/compis/Terraform_Compis2/antlr/program$ python3 main.py --destroy
Ejecutando: terraform destroy -auto-approve
digitalocean_droplet.web: Refreshing state... [id=510811168]

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        = "2025-07-30T16:09:25Z" -> null
  - disk              = 10 -> null
  - graceful_shutdown = false -> null
  - id                = "510811168" -> null
  - image             = "ubuntu-24-10-x64" -> null
  - ipv4_address       = "134.209.117.153" -> 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:510811168" -> 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 = "134.209.117.153" -> null
digitalocean_droplet.web: Destroying... [id=510811168]
digitalocean_droplet.web: Still destroying... [id=510811168, 00m10s elapsed]
digitalocean_droplet.web: Still destroying... [id=510811168, 00m20s elapsed]
digitalocean_droplet.web: Destruction complete after 21s

Destroy complete! Resources: 1 destroyed.
danidubon12@LAPTOP-0K0QVOST:~/compis/Terraform_Compis2/antlr/program$
```



## Creación Droplet:

```
bianca_cal@LAPTOP-K09637M7:~/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-K09637M7:~/Terraform_Compis2/antlr/program$ █
```

## Se crea archivo .json:

```
terraform > {} droplet_state.json > ...  
1  {  
2    "name": "example-droplet",  
3    "id": "510855960",  
4    "ip": "147.182.211.194"  
5  }
```

### Destruir Droplet con script:

```
bianca_cal@LAPTOP-K09637M7:~/Terraform_Compis2/antlr/program$ python main.py --destroy  
Ejecutando: destroy vía API  
Droplet destruido exitosamente vía API.  
bianca_cal@LAPTOP-K09637M7:~/Terraform_Compis2/antlr/program$
```

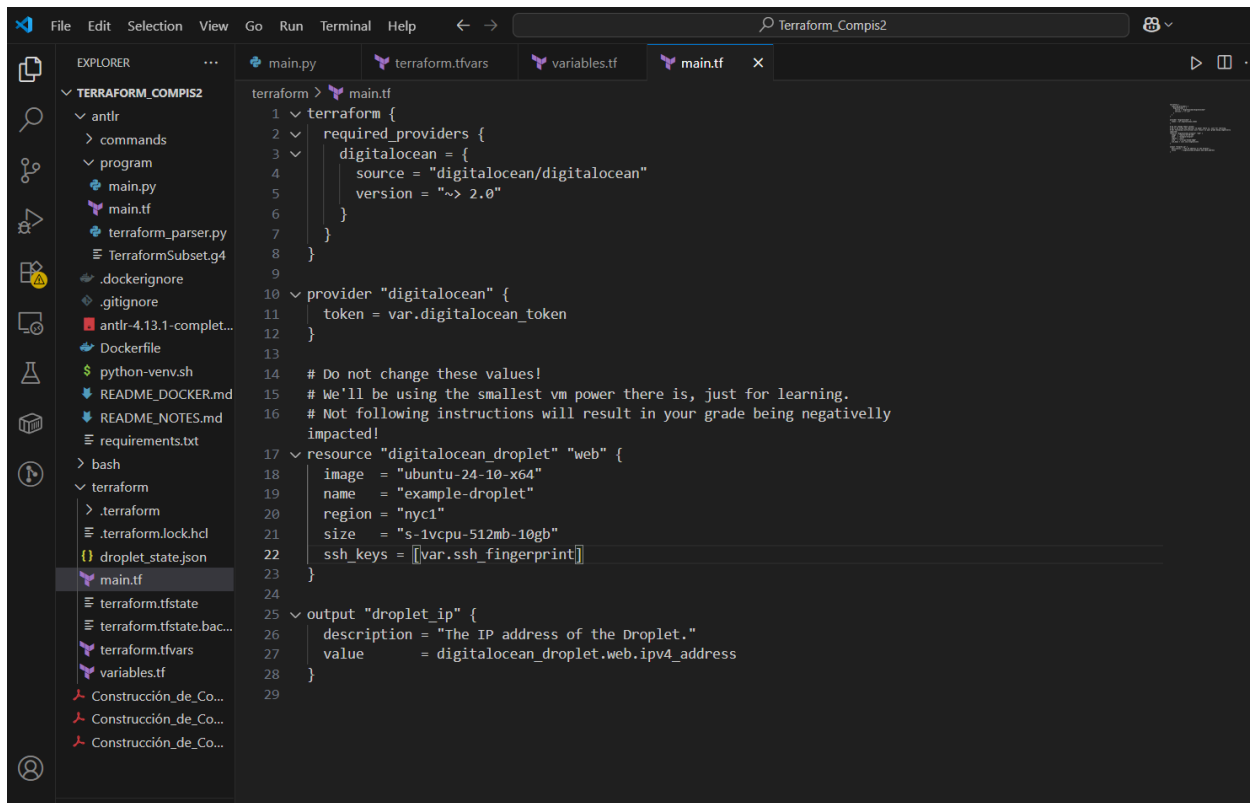
### Conexión exitosa por SSH al Droplet creado desde Terraform

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"

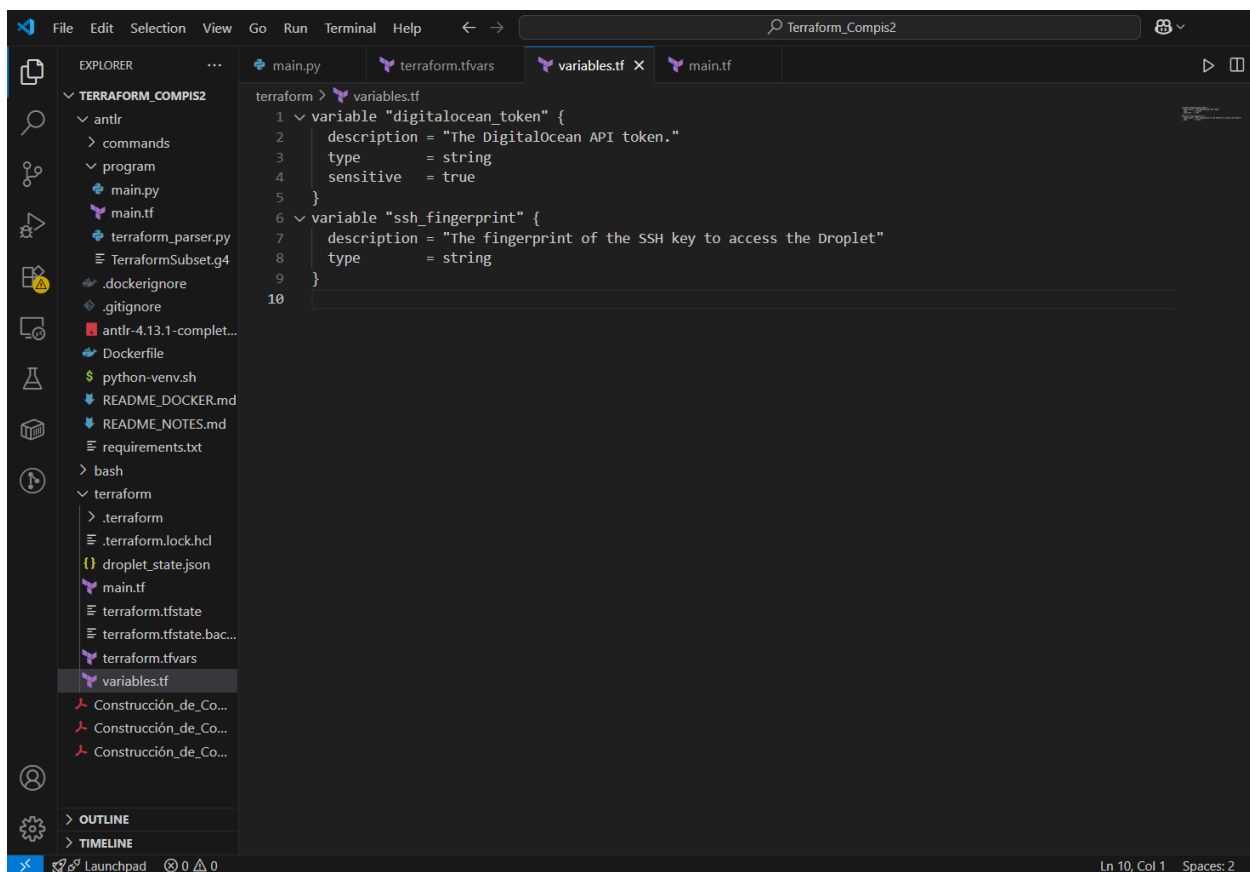
Este error indicaba que el token de acceso de la API de DigitalOcean no tenía permisos suficientes para leer las llaves SSH registradas en la cuenta.

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.



This screenshot shows the Visual Studio Code editor with a Terraform project named 'Terraform\_Compis2'. The Explorer sidebar on the left displays the project structure, including files like 'main.py', 'main.tf', 'terraform.tfvars', and 'variables.tf'. The main editor window shows the 'main.tf' file with the following Terraform configuration:

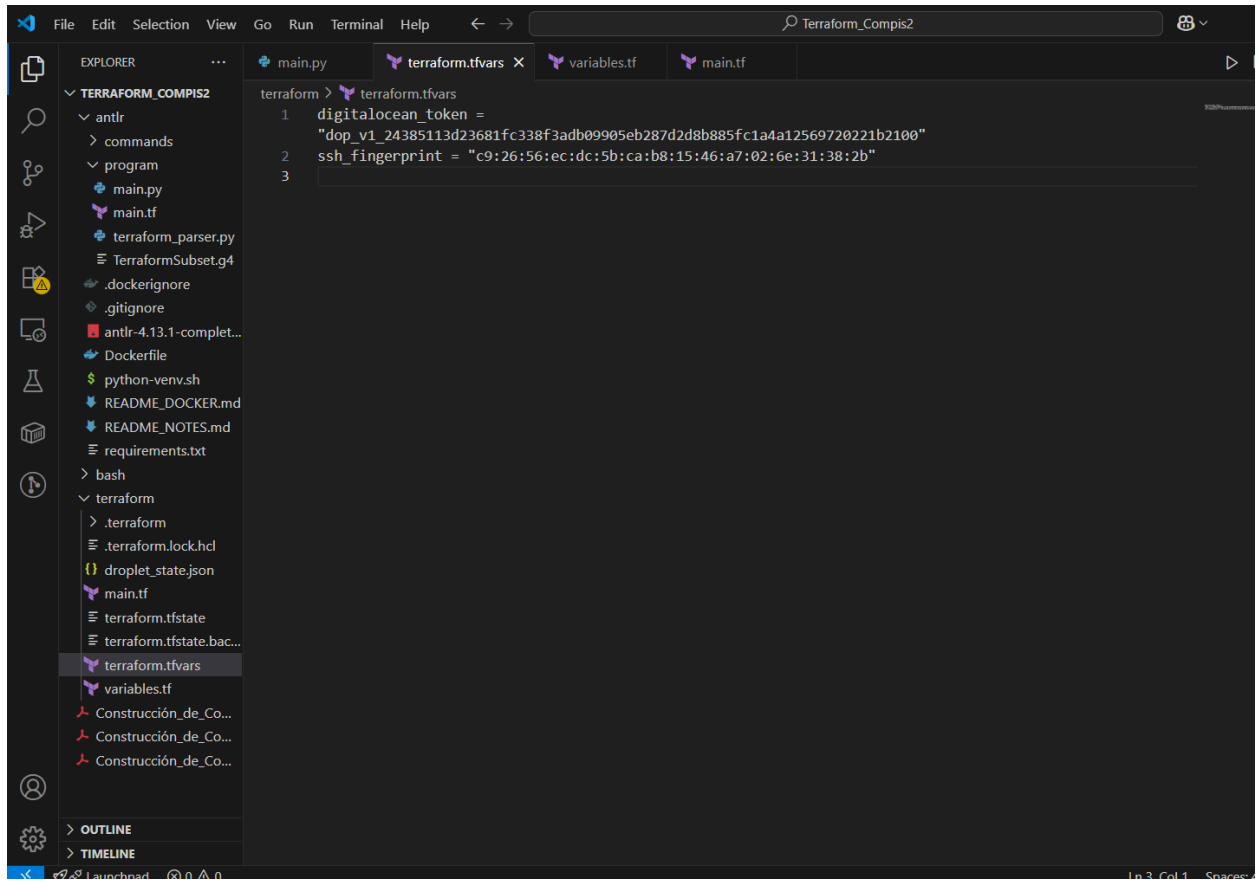
```
1 terraform {
2   required_providers {
3     digitalocean = {
4       source = "digitalocean/digitalocean"
5       version = "~> 2.0"
6     }
7   }
8 }
9
10 provider "digitalocean" {
11   token = var.digitalocean_token
12 }
13
14 # Do not change these values!
15 # We'll be using the smallest vm power there is, just for learning.
16 # Not following instructions will result in your grade being negatively
17   impacted!
18 }
19
20 resource "digitalocean_droplet" "web" {
21   image = "ubuntu-24-10-x64"
22   name = "example-droplet"
23   region = "nyc1"
24   size = "s-1vcpu-512mb-10gb"
25   ssh_keys = [var.ssh_fingerprint]
26 }
27
28 output "droplet_ip" {
29   description = "The IP address of the Droplet."
30   value = digitalocean_droplet.web.ipv4_address
31 }
```



This screenshot shows the Visual Studio Code editor with the same Terraform project. The Explorer sidebar shows the 'variables.tf' file selected. The main editor window displays the 'variables.tf' file with the following configuration:

```
1 variable "digitalocean_token" {
2   description = "The DigitalOcean API token."
3   type = string
4   sensitive = true
5 }
6
7 variable "ssh_fingerprint" {
8   description = "The fingerprint of the SSH key to access the Droplet"
9   type = string
10 }
```

The status bar at the bottom indicates the current position is 'Ln 10, Col 1' with 'Spaces: 2'.



```
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]
```

```
root@example-droplet: ~ X + v

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]
digitalocean_droplet.web: Creation complete after 36s [id=510866364]

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

Outputs:

droplet_ip = "165.227.84.16"
danidubon12@LAPTOP-0K0QVOST:~/compis/Terraform_Compis2/terraform$ ssh root@165.227.84.16
The authenticity of host '165.227.84.16 (165.227.84.16)' can't be established.
ED25519 key fingerprint is SHA256:9zvPMNbNvhn4V1JwXd3iAaHsJedZpRE0nDh/FzvS518.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '165.227.84.16' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-9-generic x86_64)

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

System information as of Wed Oct 23 18:45:02 UTC 2024

System load:   1.53      Processes:      27
Usage of /home: unknown  Users logged in: 0
Memory usage:  5%       IPv4 address for eth0: 10.10.10.2
Swap usage:    0%

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

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.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@example-droplet:~# |
```

## Enlace a video explicatorio

<https://youtu.be/ujgbcpZeA38>