# UNIVERSIDAD DEL VALLE DE GUATEMALA
## Facultad de Ingeniería

# [M1] Laboratorio 3
Uso de ANTLR para Parsear Terraform y Administrar Droplets de DigitalOcean

Sofía Velasquez, 22049
José Rodrigo Marchena, 22398
Irving Fabricio Acosta, 22781
Diego García, 22404

Gabriel Brolo Tobar
Construcción de Compiladores
Guatemala, 2025

Parte 1: Terraform

Crear con init terraform:



Ver el plan para terraform:



Apply terraform y ver la ip

```
∨ TERMINAL

● PS C:\Users\50250\Desktop\Sofía Mishell Velásquez UVG\Cuarto Año 2025\Segundo semestre\Compis\Laboratorio_3_CC\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-20-04-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-1gb"
      + 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: Still creating... [00m30s elapsed]
digitalocean_droplet.web: Creation complete after 33s [id=510626859]

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

Outputs:

droplet_ip = "174.138.36.166"
○ PS C:\Users\50250\Desktop\Sofía Mishell Velásquez UVG\Cuarto Año 2025\Segundo semestre\Compis\Laboratorio_3_CC\terraform> []
```

Destruir lo creado:

```
∨ TERMINAL                                                                              powershell - terraform  + ∨  □  🗑  …

● PS C:\Users\50250\Desktop\Sofía Mishell Velásquez UVG\Cuarto Año 2025\Segundo semestre\Compis\Laboratorio_3_CC\terraform> terraform destroy
digitalocean_droplet.web: Refreshing state... [id=510626859]

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           = "2025-07-29T15:09:542" -> null
      - disk                 = 25 -> null
      - graceful_shutdown    = false -> null
      - id                   = "510626859" -> null
      - image                = "ubuntu-20-04-x64" -> null
      - ipv4_address         = "174.138.36.166" -> null
      - ipv4_address_private = "10.116.0.22" -> null
      - ipv6                 = false -> null
      - locked               = false -> null
      - memory               = 1024 -> null
      - monitoring           = false -> null
      - name                 = "example-droplet" -> null
      - price_hourly         = 0.00893 -> null
      - price_monthly        = 6 -> null
      - private_networking   = true -> null
      - region               = "nyc1" -> null
      - resize_disk          = true -> null
      - size                 = "s-1vcpu-1gb" -> null
      - status               = "active" -> null
      - tags                 = [] -> null
      - urn                  = "do:droplet:510626859" -> null
      - private_networking   = true -> null
      - region               = "nyc1" -> null
      - resize_disk          = true -> null
      - size                 = "s-1vcpu-1gb" -> null
      - status               = "active" -> null
      - tags                 = [] -> null
      - urn                  = "do:droplet:510626859" -> 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 = "174.138.36.166" -> 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=510626859]
digitalocean_droplet.web: Still destroying... [id=510626859, 00m10s elapsed]
digitalocean_droplet.web: Still destroying... [id=510626859, 00m20s elapsed]
digitalocean_droplet.web: Destruction complete after 22s

Destroy complete! Resources: 1 destroyed.
```

## Parte 2: Bash

### Construir la imagen de Docker



### Ejecutar el script para crear el droplet



## Parte 3: ANTLR

• Compile el archivo de Terraform en la carpeta de antlr usando el driver de Python que se proporciona y vea c´omo se crea el droplet (el c´odigo imprime la IP del droplet y el ID del mismo)

```
appuser@a071e6d71c97:/program$ python3 terraform_parser.py main.tf create
[var] digitalocean_token = dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b
[*] Creating droplet...
[+] Droplet created with ID: 510898393
[*] Waiting for droplet to become active and assigned an IP...
[+] State saved to terraform.tfstate
[✓] Droplet available at IP: 157.230.48.26
```

En efecto, esta la ip 67.205.175.202 y la id de Droplet de 510683037

• Luego de hacerlo, haga un ping a la IP y vea que si est´a up el recurso

```
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC> ping 157.230.48.26

Pinging 157.230.48.26 with 32 bytes of data:
Reply from 157.230.48.26: bytes=32 time=113ms TTL=47
Reply from 157.230.48.26: bytes=32 time=95ms TTL=47
Reply from 157.230.48.26: bytes=32 time=115ms TTL=47
Reply from 157.230.48.26: bytes=32 time=101ms TTL=47

Ping statistics for 157.230.48.26:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 95ms, Maximum = 115ms, Average = 106ms
```

• Luego, haga un delete manual de este droplet usando el script de bash que us´o ya anteriormente cuando cre´o instancias con bash y la REST API. DEBE HACER DELETE MANUAL.

Primero se importo el recurso con el droplet ID

```
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC\bash> docker compose run digitalocean destroy_droplet.sh
time="2025-07-30T23:11:09-06:00" level=warning msg="C:\\Users\\JM\\Documents\\Compi2\\Laboratorio_3_CC\\bash\\docker-compose.y
ml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
time="2025-07-30T23:11:10-06:00" level=warning msg="Found orphan containers ([bash-digitalocean-run-9c64e08c53be bash-digitalo
cean-run-2863ed7a765f bash-digitalocean-run-66832a800b3d bash-digitalocean-run-758d809b48ff bash-digitalocean-run-b8ff991fa4ab
 bash-digitalocean-run-242927c54695 bash-digitalocean-run-a10b78116710 bash-digitalocean-run-7a4e37e053b1 bash-digitalocean-ru
n-c175271cde4c bash-digitalocean-run-90c7947b0747 bash-digitalocean-run-bc78101208af]) for this project. If you removed or ren
amed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up."
Droplet with ID 510898393 has been destroyed
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC\bash>
```

```
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC> ping 157.230.48.26

Pinging 157.230.48.26 with 32 bytes of data:
Request timed out.
Request timed out.

Ping statistics for 157.230.48.26:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),
Control-C
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC> clear
```

Para luego poder hacer terraform destroy, y podemos ver que la ip y droplet id indicadas ya fueron liberadas.

```
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC\terraform> terraform destroy
digitalocean_droplet.web: Refreshing state... [id=510683037]

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           = "2025-07-29T23:41:06Z" -> null
      - disk                 = 10 -> null
      - id                   = "510683037" -> null
      - image                = "ubuntu-24-10-x64" -> null
      - ipv4_address         = "67.205.175.202" -> 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:510683037" -> null
      - vcpus                = 1 -> null
      - volume_ids           = [] -> null
      - vpc_uuid             = "2ddc24ca-204a-46fe-8b73-06ce664c2cb5" -> null
```

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

Changes to Outputs:
  - droplet_ip = "67.205.175.202" -> 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=510683037]
digitalocean_droplet.web: Still destroying... [id=510683037, 00m10s elapsed]
digitalocean_droplet.web: Still destroying... [id=510683037, 00m20s elapsed]
digitalocean_droplet.web: Destruction complete after 22s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\JM\Documents\Compi2\Laboratorio_3_CC\terraform>
```

• Luego, deberá de completar el ecosistema de su propio Terraform, este archivo como está hace mimic de un ¨ terraform apply–auto-approve¨ . Entonces lo que ahora necesito que haga, es que cree el c´odigo que soporte hacer un ¨ terraform destroy–auto-approve". Para lograr esto, deberá soportar parámetros en el archivo de Python. Dependiendo del flag proporcionado al cli de Python entonces ya hará un apply o destroy.

```
PS C:\Users\dgv31\OneDrive\Documents\Universidad\Semestre 8\compis\Laboratorio_3_CC\antlr> docker run --rm -it -v "${PWD}/program:/program"
lab3-image
>>
appuser@37eca842a7b3:/program$ antlr -Dlanguage = Python3 TerraformSubset.g4
error(9):  invalid -Dname=value syntax: -Dlanguage
error(7):  cannot find or open file: =
error(7):  cannot find or open file: Python3
appuser@37eca842a7b3:/program$ antlr -Dlanguage=Python3 TerraformSubset.g4
appuser@37eca842a7b3:/program$ python3 .\terraform_parser.py .\main.tf create
python3: can't open file '/program/.terraform_parser.py': [Errno 2] No such file or directory
appuser@37eca842a7b3:/program$ python3 terraform_parser.py main.tf create
[var] digitalocean_token = dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b
[*] Creating droplet...
[+] Droplet created with ID: 510863592
[*] Waiting for droplet to become active and assigned an IP...
[+] State saved to terraform.tfstate
[√] Droplet available at IP: 68.183.96.87
appuser@37eca842a7b3:/program$ 
```

• Después, necesita crear el código para hacer un destroy, que básicamente es llamar a la REST API también. Pero para poder hacer un destroy, necesita saber el ID del droplet, entonces, deberá agregar código para crear un terraform statefile (.tfstate).

```
appuser@37eca842a7b3:/program$ python3 terraform_parser.py main.tf destroy
[var] digitalocean_token = dop_v1_46ea0abd61861407db60b9aea7510669ae34d16b179e4bde51c76369a82d356b
[+] Destroying Droplet with ID: 510863592
[+] Droplet succesfully destroyed
```

• Este statefile se debe crear cuando haga un apply. Se guardará un archivo y este tendrá un json con la información correspondiente, es decir, para el droplet con este nombre proporcionado, se asocia este ID y esta IP. Ya con este archivo, cuando haga un destroy entonces ya puede mandar el ID para destruir el recurso.

```
1    {
2      "id": 510863592,
3      "name": "example-droplet",
4      "region": {
5        "name": "New York 1",
6        "slug": "nyc1",
7        "features": [
8          "backups",
9          "ipv6",
10         "metadata",
11         "install_agent",
12         "storage",
13         "image_transfer"
14       ],
15       "available": true,
16       "sizes": [
17         "s-1vcpu-512mb-10gb",
18         "s-1vcpu-1gb",
19         "s-1vcpu-1gb-amd",
20         "s-1vcpu-1gb-intel",
21         "s-1vcpu-1gb-35gb-intel",
22         "s-1vcpu-2gb",
23         "s-1vcpu-2gb-amd",
24         "s-1vcpu-2gb-intel",
25         "s-1vcpu-2gb-70gb-intel",
26         "s-2vcpu-2gb",
27         "s-2vcpu-2gb-amd",
28         "s-2vcpu-2gb-intel",
29         "s-2vcpu-2gb-90gb-intel",
30         "s-2vcpu-4gb",
31         "s-2vcpu-4gb-amd",
32         "s-2vcpu-4gb-intel",
33         "s-2vcpu-4gb-120gb-intel",
34         "s-2vcpu-8gb-amd",
35         "c-2",
36         "c2-2vcpu-4gb",
37         "s-2vcpu-8gb-160gb-intel",
38         "s-4vcpu-8gb",
39         "s-4vcpu-8gb-amd",
40         "s-4vcpu-8gb-intel",
```