

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Compiladores 2 – Gabriel Brolo



Laboratorio 3

Daniel Alfredo Rayo Roldán, 22933

Flavio André Galán Donis, 22386

José Alejandro Prince Martinez,z 22087

Repositorio: <https://github.com/DanielRasho/Compiladore-lab3>

Video: <https://youtu.be/IEjQvq9lyNQ>

Terraform

Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working with Terraform immediately by creating Terraform configuration files.

```
[smaug@toaster lab3]$ cd terraform/
```

```
[smaug@toaster terraform]$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding digitalocean/digitalocean versions matching "~> 2.0"...

- Installing digitalocean/digitalocean v2.61.0...

- Installed digitalocean/digitalocean v2.61.0 (signed by a HashiCorp partner, key ID F82037E524B9C0E8)

Partner and community providers are signed by their developers.

If you'd like to know more about provider signing, you can read about it here:

<https://developer.hashicorp.com/terraform/cli/plugins/signing>

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.

```
[smaug@toaster terraform]$ 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:

```
# digitalocean_droplet.web will be created
+ resource "digitalocean_droplet" "web" {
  + backups      = false
```

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.

```
[smaug@toaster terraform]$ 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:

```
# 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:

```
}

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=510336535]

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

Outputs:
droplet_ip = "68.183.23.127"

[smaug@toaster terraform]$ ping 68.183.23.127
PING 68.183.23.127 (68.183.23.127) 56(84) bytes of data.
64 bytes from 68.183.23.127: icmp_seq=2 ttl=41 time=74.5 ms
64 bytes from 68.183.23.127: icmp_seq=3 ttl=41 time=75.7 ms
^C
--- 68.183.23.127 ping statistics ---
3 packets transmitted, 2 received, 33.333% packet loss, time 2010ms
rtt min/avg/max/mdev = 74.532/75.135/75.738/0.603 ms
[smaug@toaster terraform]$
```

En el último recuadro marcado de rojo se puede observar que el droplet ya puede ser “pingueado”.

```
- size           = "s-1vcpu-512mb-10gb" -> null
- status         = "active" -> null
- tags           = [] -> null
- urn            = "do:droplet:510336535" -> null
- vcpus          = 1 -> null
- volume_ids     = [] -> null
- vpc_uuid       = "aa699178-3774-44e3-ae29-533d4fc1fbd6" -> null
# (1 unchanged attribute hidden)
}

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

Changes to Outputs:
  - droplet_ip = "68.183.23.127" -> 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=510336535]
digitalocean_droplet.web: Still destroying... [id=510336535, 00m10s elapsed]
digitalocean_droplet.web: Still destroying... [id=510336535, 00m20s elapsed]
digitalocean_droplet.web: Destruction complete after 22s

Destroy complete! Resources: 1 destroyed.
```

Bash

Para la gestión de dependencias se utilizó Nix en vez de Docker, a continuación se encuentra el archivo de configuración.

```
flake.nix
26 {
25   description = "A basic multiplatform flake";
24
23   inputs = {
22     nixpkgs.url = "github:nixos/nixpkgs?ref=nixos-unstable";
21   };
20
19   outputs = {
18     self,
17     nixpkgs,
16   }: let
15     # System types to support.
14     supportedSystems = ["x86_64-linux" "x86_64-darwin" "aarch64-linux" "aarch64-darwin"];
13
12     # Helper function to generate an attrset '{ x86_64-linux = f "x86_64-linux"; ... }'.
11     forAllSystems = nixpkgs.lib.genAttrs supportedSystems;
10
9     # Nixpkgs instantiated for supported system types.
8     nixpkgsFor = forAllSystems (system: import nixpkgs {inherit system; config.allowUnfree = true;});
7   in {
6     devShells = forAllSystems (system: let
5       pkgs = nixpkgsFor.${system};
4       in {
3         default = pkgs.mkShell {
2           packages = [
1             pkgs.terraform
27             pkgs.jq
1             ];
1         };
2       };
3     };
4   };
5 }
6
```

A continuación se muestran los 2 comandos para la creación y destrucción de un droplet ejecutados con éxito.

```
[smaug@toaster ~]$ cd /root/.nix-profile/bin/
[smaug@toaster ~]$ ./create_droplet.sh
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 3975 100 3878 100  97 3539    88 0:00:01 0:00:01 --:--:-- 3626
Droplet created with ID: 510337886
[smaug@toaster ~]$ ./destroy_droplet.sh
Droplet with ID 510337886 has been destroyed
[smaug@toaster ~]$
```

< 0 0 Live Share AWS: profile:default 879 hours 11 minutes

ANTLR

Creacion del droplet

```
12
11 print("[*] Waiting for droplet to become active and assigned an IP...")
10 while True:
9     resp = requests.get(f"https://api.digitalocean.com/v2/droplets/{droplet_id}", headers=headers)
8     droplet_info = resp.json()["droplet"]
7     networks = droplet_info["networks"]["v4"]
6     public_ips = [n["ip_address"] for n in networks if n["type"] == "public"]
5     if public_ips:
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

[*] Creating droplet...
Traceback (most recent call last):
File "/home/smaug/Documents/COMPILADORES/antlr/program/terraform_parser.py", line 114, in <module>
main(sys.argv)
File "/home/smaug/Documents/COMPILADORES/antlr/program/terraform_parser.py", line 110, in main
ip = create_droplet(token, listener.droplet_config)
File "/home/smaug/Documents/COMPILADORES/antlr/program/terraform_parser.py", line 80, in create_droplet
response.raise_for_status()
File "/home/smaug/Documents/COMPILADORES/.venv/lib/python3.13/site-packages/requests/models.py", line 1026, in raise_for_status
raise HTTPError(http_error_msg, response=self)
requests.exceptions.HTTPError: 401 Client Error: Unauthorized for url: https://api.digitalocean.com/v2/droplets

```
~/COMPILADORES/antlr/program main v3.13.5 16:18 python terraform_parser.py main.tf
[var] digitalocean_token = dop_v1_1486681a5d0a0a84b6f3f15b75a4df85999c69e1695e2a106033518afa281b94
[*] Creating droplet...
[+] Droplet created with ID: 510676364
[*] Waiting for droplet to become active and assigned an IP...
[✓] Droplet available at IP: 68.183.117.195
```

~/COMPILADORES/antlr/program main v3.13.5 16:19

~/COMPILADORES/antlr/program main v3.13.5 16:19

main* 0 1 Live Share AWS: profile:default 887 hours 39 minutes DanielRasho (23 hours ago) Ln 96, Col

El droplet creado ya se muestra en el dashboard


Search by resource name or public IP (Ctrl+B) Create ? 🔔 My Te
Estimated costs: \$0

Droplets

Create an Autoscale Pool Create Droplet

Droplets Autoscale Pools

Search by Droplet name

Name	IP Address	Created	Tags
 example-droplet 512 MB / 10 GB Disk / NYC1 - Ubuntu 24.10 x64			Upsize More

Ping al droplet creado

```
requests.exceptions.HTTPError: 401 Client Error: Unauthorized for url: https://api.digitalocean.com/v2/droplets

~/COMPILADORES/antlr/program main v3.13.5 16:18 python terraform_parser.py main.tf
[var] digitalocean_token = dop_v1_1486681a5d0a0a84b6f3f15b75a4df85999c69e1695e2a106033518afa281b94
[*] Creating droplet...
[+] Droplet created with ID: 510676364
[*] Waiting for droplet to become active and assigned an IP...
[✓] Droplet available at IP: 68.183.117.195

~/COMPILADORES/antlr/program main v3.13.5 16:19
PING 68.183.117.195 (68.183.117.195) 56(84) bytes of data.
64 bytes from 68.183.117.195: icmp_seq=1 ttl=51 time=65.3 ms
64 bytes from 68.183.117.195: icmp_seq=2 ttl=51 time=64.7 ms
64 bytes from 68.183.117.195: icmp_seq=3 ttl=51 time=64.8 ms
64 bytes from 68.183.117.195: icmp_seq=4 ttl=51 time=65.5 ms
```

Eliminación del droplet manualmente

```
--- 68.183.117.195 ping statistics ---
63 packets transmitted, 63 received, 0% packet loss, time 62089ms
rtt min/avg/max/mdev = 61.721/65.256/75.548/3.197 ms

~/COMPILADORES/antlr/program main v3.13.5 16:22 cd ..

~/COMPILADORES/antlr main v24.0.2 v3.13.5 16:22 export DO_API_TOKEN="dop_v1_1486681a5d0a0a84b6f3f15b75a4df85999c69e1695e2a106033518afa281b94"

~/COMPILADORES/antlr main v24.0.2 v3.13.5 16:23 nix develop
path '/home/smaug/Documents/COMPILADORES/antlr' does not contain a 'flake.nix', searching up
warning: Git tree '/home/smaug/Documents/COMPILADORES' is dirty
[smaug@toaster antlr]$ cd ..
[smaug@toaster COMPILADORES]$ cd bash/
[smaug@toaster bash]$ ./destroy_droplet.sh
Droplet with ID 510676364 has been destroyed
[smaug@toaster bash]$
```

Destrucción de Recursos en la nube

```
[elrohirgt@elrohirgt:~/Documents/Development/Compiladore-lab3/antlr/program]$ ./terraform_parser.py main.tf
[var] digitalocean_token = dop_v1_1486681a5d0a0a84b6f3f15b75a4df85999c69e1695e2a106033518afa281b94
[*] Creating droplet...
[+] Droplet created with ID: 510684065
[*] Waiting for droplet to become active and assigned an IP...
[✓] Droplet available at IP (64.227.13.229) with ID (510684065)

[elrohirgt@elrohirgt:~/Documents/Development/Compiladore-lab3/antlr/program]$ ./terraform_parser.py -d main.tf
[var] digitalocean_token = dop_v1_1486681a5d0a0a84b6f3f15b75a4df85999c69e1695e2a106033518afa281b94
[*] Deleting droplet...
[+] Droplet deleted with ID: 510684065
[✓] Resources deleted!

[elrohirgt@elrohirgt:~/Documents/Development/Compiladore-lab3/antlr/program]$
```

Configuración SSH Key

La configuración de la ssh key se realiza durante la creación del droplet, se verifica si la key usada ya existe o no. Caso de ssh key existente:

```
(venv) [akice@Akice-PC program]$ python terraform_parser.py main.tf
[var] digitalocean_token = dop_v1_6342c964d1100c1e96f09fccc575574c314588939136efc413d8c874a2587654
[i] SSH key already exists in DigitalOcean: prince (1a:87:ab:a0:2a:46:7c:db:08:fe:c1:d1:93:6b:82:79)
[*] Creating droplet...
[+] Droplet created with ID: 510819538
[*] Waiting for droplet to become active and assigned an IP...
[✓] Droplet available at IP (134.209.117.224) with ID (510819538)
```

En el caso de querer usar un ssh key nueva, primero se debe subir a digital ocean porque de no subirla, da error 403 indicando que el token no tiene los permisos para realizar la acción de subir la ssh key a digital ocean:

```
[var] digitalocean_token = dop_v1_6342c964d1100c1e96f09fccc575574c314588939136efc413d8c874a2587654
Traceback (most recent call last):
  File "/home/akice/Documents/Repositories/Compiladore-lab3/antlr/program/terraform_parser.py", line 220,
in <module>
    main()
    ~~~~^
  File "/home/akice/Documents/Repositories/Compiladore-lab3/antlr/program/terraform_parser.py", line 213,
in main
    (ip, id) = create_droplet(token, listener.droplet_config)
               ~~~~~^~~~~~
  File "/home/akice/Documents/Repositories/Compiladore-lab3/antlr/program/terraform_parser.py", line 137,
in create_droplet
    add_resp.raise_for_status()
    ~~~~~^
  File "/home/akice/Documents/Repositories/Compiladore-lab3/antlr/venv/lib/python3.13/site-packages/requests/models.py", line 1026, in raise_for_status
    raise HTTPError(http_error_msg, response=self)
requests.exceptions.HTTPError: 403 Client Error: Forbidden for url: https://api.digitalocean.com/v2/account/keys
```