



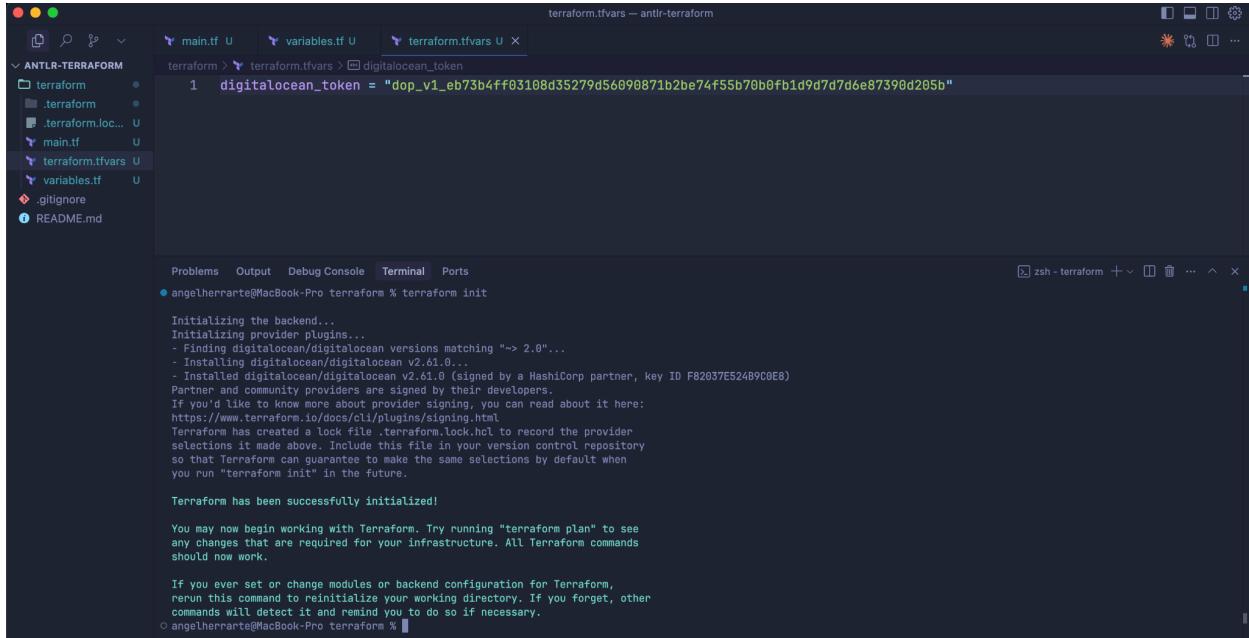
Laboratorio 3: Antlr - Terraform - DigitalOcean

Construcción de Compiladores

Angel Andres Herrarte Lorenzana, 22873

27 de Julio del 2025

Fase 1: Terraform



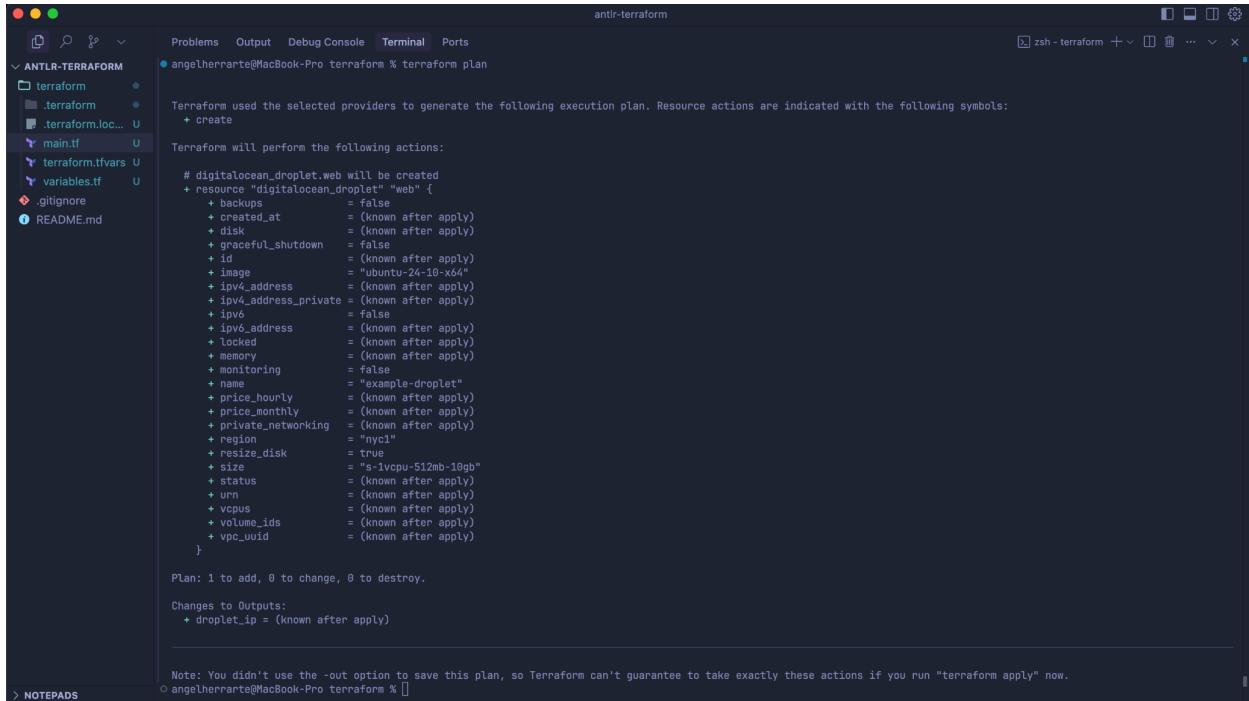
```
terraforming the backend...
Initializing provider plugins...
- Finding digitalocean/digitalocean versions matching "~> 2.0"...
- Installing digitalocean/digitalocean v2.61.0...
- Installed digitalocean/digitalocean v2.61.0 (signed by a HashiCorp partner, key ID F82037E524B9C0E8)
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/ci/plugins/signing.html
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.

○ angelherrarte@MacBook-Pro terraform %
```



```
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)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

○ angelherrarte@MacBook-Pro terraform %
```

ANTLR-TERRAFORM

- terraform
 - .terraform
 - .terraform.lock.hcl
 - main.tf
 - terraform.tfstate
 - terraform.tfvars
 - variables.tf
 - .gitignore
 - README.md

Problems Output Debug Console **Terminal** Ports

```
+ 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... [10s elapsed]
digitalocean_droplet.web: Still creating... [20s elapsed]
digitalocean_droplet.web: Creation complete after 23s [id=510285239]

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

Outputs:
```

> NOTEPADS > OUTLINE

zsh - terraform

angelherrarte@MacBook-Pro terraform %

ANTLR-TERRAFORM

- terraform
 - .terraform
 - .terraform.lock.hcl
 - main.tf
 - terraform.tfstate
 - terraform.tfvars
 - variables.tf
 - .gitignore
 - README.md

Problems Output Debug Console **Terminal** Ports

```
- status          = "active" -> null
- tags           = [] -> null
- urn            = "do:droplet:510285239" -> 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 = "167.99.155.199" -> 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=510285239]
digitalocean_droplet.web: Still destroying... [id=510285239, 10s elapsed]
digitalocean_droplet.web: Still destroying... [id=510285239, 20s elapsed]
digitalocean_droplet.web: Destruction complete after 23s

Destroy complete! Resources: 1 destroyed.
```

> NOTEPADS > OUTLINE

zsh - terraform

angelherrarte@MacBook-Pro terraform %

Fase 2: Bash

The screenshot shows the VS Code interface with the title bar "antlr-terraform". The left sidebar shows a file tree for a project named "ANTLR...". The terminal tab is active, displaying the command "zsh - bash". The output pane shows the following log:

```
=> sha256:8fcb4d8ca5354def3d8fc243717141ce31e2c428791f6682bd2fafef15388214 6.69kB / 6.69kB
=> sha256:722ea796ac2d57eeb3627c58a582fc1acc58ba51faf815e1bce1682ae5c092f7 424B / 424B
=> sha256:674fd86895ad5fffb4eep9b0fffb45a993e135df8ff9eac5c956460f395a8f540d9d 2.31kB / 2.31kB
=> sha256:ecd83b6c354452b6a99797c7666bb1a927f1e0e02afbf6e401dd6f87d5db8576 25.98MB / 25.98MB
=> extracting sha256:ecd3b6c354452b6a99797c7666bb1a927f1e0e02afbf6e401dd6f87d5db8576
[DigitalOcean internal] load build context
=> transferring context: 1.89kB
=> [DigitalOcean 2/6] RUN apt-get update && apt-get install -y curl jq && rm -rf /var/lib/apt/lists/*
=> [DigitalOcean 3/6] COPY create_droplet.sh /usr/local/bin/create_droplet.sh
=> [DigitalOcean 4/6] COPY destroy_droplet.sh /usr/local/bin/destroy_droplet.sh
=> [DigitalOcean 5/6] RUN chmod +x /usr/local/bin/create_droplet.sh
=> [DigitalOcean 6/6] RUN chmod +x /usr/local/bin/destroy_droplet.sh
=> [DigitalOcean] exporting to image
=> exporting layers
=> writing image sha256:7efb60493c16a0def8bcd9e4275f64471b22f8535131f57dff35cef19d869
[+] Building 1/1 docker.io/library/bash-digitalocean
  ✓ Service digitalocean Built
angelherrarte@MacBook-Pro bash %
```

The bottom status bar shows "zsh - bash" and the current time.

The screenshot shows the VS Code interface with the title bar "droplet_id.txt - antlr-terraform-digitalocean". The left sidebar shows a file tree for a project named "ANTLR-TERRAFORM-DL...". The terminal tab is active, displaying the command "zsh - bash". The output pane shows the following log:

```
bash > droplet_id.txt
1 | 6987298
2
```

Below the log, there is a warning message from Docker Compose:

```
WARN[0000] /Users/angelherrarte/Desktop/antlr-terraform-digitalocean/docker-compose.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] Found orphan containers ([bash-digitalocean-run-f88d66c695b2 bash-digitalocean-run-457a3b856590 bash-digitalocean-run-7d2c8be1f18d bash-digitalocean-run-5c08af539ce7 bash-digitalocean-run-db3ee988a54b]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload Upload Total Spent   Left  Speed
100  161  100    64  100   97  139  210 --:--:-- --:--:-- 349
Droplet created with ID: 6987298
angelherrarte@MacBook-Pro bash %
```

The bottom status bar shows "zsh - bash" and the current time.

```

create_droplet.sh M droplet_id.txt M destroy_droplet.sh M docker-compose.yaml M
bash > droplet_id.txt
1 | 6987298
2 | %L to chat, %K to generate

Problems Output Debug Console Terminal Ports
● angelherrarte@MacBook-Pro bash % docker-compose run digitalocean /usr/local/bin/destroy_droplet.sh

WARN[0800] /Users/angelherrarte/Desktop/antlr-terraform-digitalocean/bash/docker-compose.yaml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0800] Found orphan containers ([bash-digitalocean-run-8e6977badac9 bash-digitalocean-run-dd26f39dc532 bash-digitalocean-run-75f1645a8b2b bash-digitalocean-run-914bbff58d7c bash-digitalocean-run-f88d6c69502 bash-digitalocean-run-457a3b0856590 bash-digitalocean-run-7d2c8be1f18d bash-digitalocean-run-5cd08af539ce7 bash-digitalocean-run-db3ee988a54b]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.

> NOTEPADS Droplet with ID 6987298 has been destroyed
> OUTLINE ○ angelherrarte@MacBook-Pro bash %

```

Fase 3: ANTLR

```

main.tf M
main > program > main.tf > variable "digitalocean_token" > type

13 }
14
15 # Remember to change to your PAT
variable "digitalocean_token" {
16   description = "The DigitalOcean API token."
17   type        = string
18   sensitive   = true
19   default     = "dop_v1_eb73b4ff03108d35279d56090871b2be74f55b70b0fb1d9d7d6e87390d205b"
20 }
21
22

Problems Output Debug Console Terminal Ports
⇒ [ 7/28] RUN chmod +x /usr/local/bin/antlr
⇒ [ 8/28] COPY ./commands/antlr /usr/bin/antlr
⇒ [ 9/28] RUN chmod +x /usr/bin/antlr
⇒ [10/28] COPY ./commands/grun /usr/local/bin/grun
⇒ [11/28] RUN chmod +x /usr/local/bin/grun
⇒ [12/28] COPY ./commands/grun /usr/bin/grun
⇒ [13/28] RUN chmod +x /usr/bin/grun
⇒ [14/28] COPY python-venv.sh .
⇒ [15/28] RUN chmod +x ./python-venv.sh
⇒ [16/28] RUN ./python-venv.sh
⇒ [17/28] COPY requirements.txt .
⇒ [18/28] RUN pip install -r requirements.txt --break-system-packages
⇒ [19/28] RUN adduser --disabled-password --gecos "" --home "$(pwd)" --no-create-home --uid "1001" "appuser"
⇒ [20/28] WORKDIR /program
⇒ exporting to image
⇒ => exporting layers
⇒ => writing image sha256:e7e888a7e589a59b1e41d30c3371e7226caae002e402febbb3c9e52dd2f162c
⇒ => naming to docker.io/library/antlr-terraform
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/orf3ok9niyhfnyta15wcqm

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview
○ angelherrarte@MacBook-Pro antlr %

```

```

main.tf — antlr-terraform-digitalocean
main.tf M ✘
antlr > program > main.tf > variable "digitalocean_token" > type
13 }
14
15 # Remember to change to your PAT
variable "digitalocean_token" {
  description = "The DigitalOcean API token."
  type        = string
  sensitive   = true
  default     = "dop_v1_eb73b4ff03108d35279d56090871b2be74f55b70b0fb1d9d7d6e87390d205b"
}

Problems Output Debug Console Terminal Ports
--> CACHED [11/28] RUN chmod +x /usr/local/bin/grun
--> CACHED [12/28] COPY ./commands/grun /usr/bin/grun
--> CACHED [13/28] RUN chmod +x /usr/bin/grun
--> CACHED [14/28] COPY python-venv.sh .
--> CACHED [15/28] RUN chmod +x ./python-venv.sh
--> CACHED [16/28] RUN ./python-venv.sh
--> CACHED [17/28] COPY requirements.txt .
--> CACHED [18/28] RUN pip install -r requirements.txt --break-system-packages
--> CACHED [19/28] RUN adduser --disabled-password --gecos "" --home "${pwd}" --no-create-home --uid "1001" "appuser"
--> CACHED [20/28] WORKDIR /program
--> exporting to image
--> exporting layers
--> writing image sha256:e7e888a7e589a59b11e41d30c3371e7226caae002e402febbb3c9e52dd2f162c
--> naming to docker.io/library/antlr-terraform
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/oenv6v8ynl3e5yu03bftwkug

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview
● angeherrarte@MacBook-Pro antlr % cd /program
cd: no such file or directory: /program
○ angeherrarte@MacBook-Pro antlr % docker run -it -v ${pwd}/program:/program antlr-terraform bash
appuser@3c6cd590b61a:/program$ █

```

```

main.tf — antlr-terraform-digitalocean
main.tf M ✘
antlr > program > main.tf > variable "digitalocean_token" > type
13 }
14
15 # Remember to change to your PAT
variable "digitalocean_token" {
  description = "The DigitalOcean API token."
  type        = string
  sensitive   = true
  default     = "dop_v1_eb73b4ff03108d35279d56090871b2be74f55b70b0fb1d9d7d6e87390d205b"
}

Problems Output Debug Console Terminal Ports
--> CACHED [19/28] RUN adduser --disabled-password --gecos "" --home "${pwd}" --no-create-home --uid "1001" "appuser"
--> CACHED [20/28] WORKDIR /program
--> exporting to image
--> exporting layers
--> writing image sha256:e7e888a7e589a59b11e41d30c3371e7226caae002e402febbb3c9e52dd2f162c
--> naming to docker.io/library/antlr-terraform
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/oenv6v8ynl3e5yu03bftwkug

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview
● angeherrarte@MacBook-Pro antlr % cd /program
cd: no such file or directory: /program
○ angeherrarte@MacBook-Pro antlr % docker run -it -v ${pwd}/program:/program antlr-terraform bash
appuser@3c6cd590b61a:/program$ cd /program
appuser@3c6cd590b61a:/program$ antlr TerraformSubset.g4 -Dlanguage=Python3
appuser@3c6cd590b61a:/program$ █

```

main.tf — antlr-terraform-digitalocean

```

variable "digitalocean_token" {
  description = "The DigitalOcean API token."
  type        = string
  sensitive   = true
  default     = "dop_v1_545897890727010b9109f5a4a653923358f66e1d41bccb46f7b9f8ac39755cf4"
}

```

Problems Output Debug Console Terminal Ports

Terminal output:

```

angelo@MacBook-Pro antlr % docker run -it -v $(pwd)/program antlr-terraform bash
appuser@31468e9c5ec0:/program$ cd /program
appuser@31468e9c5ec0:/program$ antlr TerraformSubset.g4 -Dlanguage=Python3
appuser@31468e9c5ec0:/program$ python terraform_parser.py main.tf
bash: python: command not found
appuser@31468e9c5ec0:/program$ python3 terraform_parser.py main.tf
[+] digitalocean_token = dop_v1_545897890727010b9109f5a4a653923358f66e1d41bccb46f7b9f8ac39755cf4
[*] Creating droplet...
[*] Droplet created with ID: 510298140
[*] Waiting for droplet to become active and assigned an IP...
[+] Droplet available at IP: 67.205.156.71
appuser@31468e9c5ec0:/program$ 

```

main.tf — antlr-terraform-digitalocean

```

variable "digitalocean_token" {
  description = "The DigitalOcean API token."
  type        = string
  sensitive   = true
  default     = "dop_v1_545897890727010b9109f5a4a653923358f66e1d41bccb46f7b9f8ac39755cf4"
}

```

Problems Output Debug Console Terminal Ports

Terminal output:

```

appuser@31468e9c5ec0:/program$ cd /program
appuser@31468e9c5ec0:/program$ antlr TerraformSubset.g4 -Dlanguage=Python3
appuser@31468e9c5ec0:/program$ python terraform_parser.py main.tf
bash: python: command not found
appuser@31468e9c5ec0:/program$ python3 terraform_parser.py main.tf
[+] digitalocean_token = dop_v1_545897890727010b9109f5a4a653923358f66e1d41bccb46f7b9f8ac39755cf4
[*] Creating droplet...
[*] Droplet created with ID: 510298140
[*] Waiting for droplet to become active and assigned an IP...
[+] Droplet available at IP: 67.205.156.71
appuser@31468e9c5ec0:/program$ exit
exit
angelo@MacBook-Pro antlr % ping 67.205.156.71
PING 67.205.156.71 (67.205.156.71): 56 data bytes
64 bytes from 67.205.156.71: icmp_seq=0 ttl=42 time=75.088 ms
64 bytes from 67.205.156.71: icmp_seq=1 ttl=42 time=85.086 ms
64 bytes from 67.205.156.71: icmp_seq=2 ttl=42 time=65.562 ms
64 bytes from 67.205.156.71: icmp_seq=3 ttl=42 time=72.164 ms
64 bytes from 67.205.156.71: icmp_seq=4 ttl=42 time=66.084 ms
64 bytes from 67.205.156.71: icmp_seq=5 ttl=42 time=73.468 ms
64 bytes from 67.205.156.71: icmp_seq=6 ttl=42 time=68.445 ms
64 bytes from 67.205.156.71: icmp_seq=7 ttl=42 time=73.537 ms
64 bytes from 67.205.156.71: icmp_seq=8 ttl=42 time=72.594 ms
64 bytes from 67.205.156.71: icmp_seq=9 ttl=42 time=72.079 ms

```

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a folder named "ANTLR-TERRAFORM-DIGITALOCEAN" containing files: antlr, bash, create_droplet.sh, destroy_droplet.sh, docker-compose.yaml, Dockerfile, droplet_id.txt, terraform, .gitignore, and README.md.
- Terminal:** The terminal window shows the command "bash > droplet_id.txt" and the output "1 510298140".
- Output:** The Output tab shows the message "Droplet with ID 510298140 has been destroyed".
- Terminal:** The terminal shows the message "Droplet with ID 510298140 has been destroyed".

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the same folder structure as the first screenshot.
- Terminal:** The terminal window shows the command "python3 terraform_parser.py main.tf --apply" and the output of the Terraform apply process, including creating a droplet with ID 510298784 and waiting for it to become active.
- Output:** The Output tab shows the Terraform apply logs.
- Terminal:** The terminal shows the Terraform apply logs.

```

1  {
2      "version": 4,
3      "terraform_version": "custom",
4      "resources": [
5          {
6              "mode": "managed",
7              "type": "digitalocean_droplet",
8              "name": "web",
9              "provider": "digitalocean",
10             "instances": [
11                 {
12                     "attributes": {
13                         "id": "510298784",
14                         "name": "example-droplet",
15                         "region": "nyc1",
16                         "size": "s-1vcpu-512mb-10gb",
17                         "image": "ubuntu-24-10-x64",
18                         "ipv4_address": "67.207.92.113"
19                     }
20                 }
21             ]
22         }
23     ]
24 }

```

Problems Output Debug Console Terminal Ports

```

[*] Creating droplet...
[*] Droplet created with ID: 510298784
[*] Waiting for droplet to become active and assigned an IP...
[*] State file saved: terraform.tfstate
[*] Droplet available at IP: 67.207.92.113
[*] Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
appuser@e9034ff99b87:/program$ 

```

```

1  {
2      "version": 4,
3      "terraform_version": "custom",
4      "resources": [
5          {
6              "mode": "managed",
7              "type": "digitalocean_droplet",
8              "name": "web",
9              "provider": "digitalocean",
10             "instances": [
11                 {
12                     "attributes": {
13                         "id": "510298784",
14                         "name": "example-droplet",
15                         "region": "nyc1",
16                         "size": "s-1vcpu-512mb-10gb",
17                         "image": "ubuntu-24-10-x64",
18                         "ipv4_address": "67.207.92.113"
19                     }
20                 }
21             ]
22         }
23     ]
24 }

```

Problems Output Debug Console Terminal Ports

```

appuser@e9034ff99b87:/program$ python3 terraform_parser.py main.tf --destroy
[var] digitalocean_token = dop_v1_545697890727010b9109f5a4a653923358f66e1d41bccb46f7b9f8ac39755cf4
[*] Destroying droplet with ID: 510298784
[*] Droplet 510298784 destroyed successfully
[*] State file removed
[*] Destroy complete! Resources: 0 added, 0 changed, 1 destroyed.
appuser@e9034ff99b87:/program$ 

```

Repositorio

<https://github.com/aherrarte2019037/antlr-terraform-digitalocean>

Vídeo

<https://youtu.be/oNQDw5z7GPI>