



**Laboratorio**  
PRACTICA  
TERRAFORM



## CONTROL DE VERSIONES

<b>Elaborado por:</b> Jonatan Stiven Gutierrez	<b>No. de Versión:</b> 1.0.0
<b>Revisado por:</b>	<b>Fecha de revisión:</b>
<b>Aprobado por:</b>	<b>Fecha de Aprobación:</b>

### Historia de Modificaciones

No. de Versión	Fecha de Versión	Autor	Revisado por	Aprobado por	Descripción
1.0.0	21/02/2024	Jonatan Stiven Gutierrez			Documento Original

### Lista de distribución

Para	Acción*	Empresa	Firma/Medio de Entrega

\* Tipos de acción: Aprobar, Revisar, Informar, Archivar, Complementar, Asistir a junta, Otras (por favor especificar)



## Contenido

<b>INTRODUCCION .....</b>	<b>4</b>
<b>PRERREQUISITOS .....</b>	<b>4</b>
<b>EJERCICIO 1:.....</b>	<b>5</b>



## **INTRODUCCION**

El siguiente documento proporciona una introducción detallada a los ejercicios realizados en el laboratorio.

## **PRERREQUISITOS**

- Tener entendimiento de los temas vistos anteriormente.



## EJERCICIO 1:

En este ejercicio usaremos el Provider "local\_file".

1. Estando en consola creamos un directorio.
  - `mkdir ejercicio1`
2. Ejecutamos el siguiente comando
  - `cd ejercicio1`
3. Ejecutamos el siguiente comando
  - `code .`  
Este comando "code ." nos lleva directamente a Visual Studio Code.
4. Estando en visual Studio Code, verificamos que estamos en la ruta previamente conocida.
5. Creamos un archivo
  - Terraform.tf  
Dentro de él, copiamos el siguiente código.

```
1 resource "local_file" "cali" {
2     content = "Hola, grupo devops"
3     filename = "seti.txt"
4 }
```

6. Volvemos a la consola.
7. Ejecutamos el siguiente comando
  - `terraform init`

```
jonatangubuntu:~/practica_terraform$ cd practica_01/
jonatangubuntu:~/practica_terraform/practica_01$ code .
jonatangubuntu:~/practica_terraform/practica_01$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.4.1...
- Installed hashicorp/local v2.4.1 (signed by HashiCorp)

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.
```



## 8. Ejecutamos el siguiente comando

- terraform plan

```
jonatan@ubuntu:~/practica_terraform/practica_01$ 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:

# local_file.cali will be created
+ resource "local_file" "cali" {
+   content      = "Hola, grupo devops"
+   content_base64sha256 = (known after apply)
+   content_base64sha512 = (known after apply)
+   content_md5   = (known after apply)
+   content_sha1  = (known after apply)
+   content_sha256 = (known after apply)
+   content_sha512 = (known after apply)
+   directory_permission = "0777"
+   file_permission  = "0777"
+   filename       = "seti.txt"
+   id             = (known after apply)
}

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

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.
```

## 9. Ejecutamos el siguiente comando

- terraform apply
- En "Enter a value" escribe: **yes**
- Presiona enter

```
jonatan@ubuntu:~/practica_terraform/practica_01$ 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:

# local_file.cali will be created
+ resource "local_file" "cali" {
+   content      = "Hola, grupo devops"
+   content_base64sha256 = (known after apply)
+   content_base64sha512 = (known after apply)
+   content_md5   = (known after apply)
+   content_sha1  = (known after apply)
+   content_sha256 = (known after apply)
+   content_sha512 = (known after apply)
+   directory_permission = "0777"
+   file_permission  = "0777"
+   filename       = "seti.txt"
+   id             = (known after apply)
}

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

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
local_file.cali: Creating...
local_file.cali: Creation complete after 0s [id=eefe216098c4bffe9b9ca742d6bc6234633cb7a1]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

## 10. Ejecutamos el siguiente comando para ver los archivos del directorio

- ls-ltr

```
jonatan@ubuntu:~/practica_terraform/practica_01$ ls -ltr
total 12
-rw-rw-r-- 1 jonatan jonatan 104 feb 22 07:54 terraform.tf
-rwxrwxr-x 1 jonatan jonatan  18 feb 22 07:55 seti.txt
-rw-rw-r-- 1 jonatan jonatan 1434 feb 22 07:55 terraform.tfstate
jonatan@ubuntu:~/practica_terraform/practica_01$
```

## 11. Ejecutamos el siguiente comando para ver el contenido del archivo .txt

- cat seti.txt

```
jonatan@ubuntu:~/practica_terraform/practica_01$ cat seti.txt
Hola, grupo devopsjonatan@ubuntu:~/practica_terraform/practica_01$
```



12. Ahora vamos a editar el archivo seti.txt

- Volvemos al Visual Studio Code
- Modificamos el archivo terraform.tf

```
1 ~ resource "local_file" "cali" {
2   content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
3   filename = "seti.txt"
4 }
```

13. Una vez editado el archivo, vamos a actualizar el estado de archivo seti.txt.

- Volvemos a la consola.
- Ya no ejecutaríamos primero el comando terraform init, si no "terraform plan".
- Esto nos muestra que hubo un cambio en el contenido, pero también nos dice que el archivo fue remplazado (porque no puede editar el archivo), entonces el destruye el archivo anterior y agrega el archivo de nuevo.

```
jonatan@ubuntu:~/practica terraform/practica 01$ terraform plan
local_file.cali: Refreshing state... [id=eefe216098c4bffe9b9ca742d6bc6234633cb7a1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
/- destroy and then create replacement

Terraform will perform the following actions:

# local_file.cali must be replaced
/- resource "local_file" "cali" {
  ~ content      = "Hola, grupo devops" -> "Hola, grupo devops 2024, este es un ejemplo de terraform" # forces replacement
  ~ content_base64sha256 = "/PnI3wZGKLGeBva8zuYmVpooGKP8fJ35lyLVFDMx0oE=" -> (known after apply)
  ~ content_base64sha512 = "r3paWlUPLmLfkIz95vkkrbXV7Znoac2WpmiGGYzHFXj4ty46SR15Q0HkQLMaj4c6cqPT6GfdbFmJThRRBRw==" -> (known after apply)
  ~ content_md5      = "b22d347a9da1f833729031c1ef76f577" -> (known after apply)
  ~ content_sh1      = "eefe216098c4bffe9b9ca742d6bc6234633cb7a1" -> (known after apply)
  ~ content_sh256     = "fc9c89b75b318a2c67bcbda873b989d5a8a8e28fd1f2529722ef1433313a81" -> (known after apply)
  ~ content_sha512    = "af7a5a588b8f2e69459258b3f52be492b6d757b667a1a7365a99a21866331c5e3e2dc8be924754903879102d66a3e1ce9ca8f4fa19f75b14c6e34e1f11441af" -> (known after apply)
  ~ id              = "eefe216098c4bffe9b9ca742d6bc6234633cb7a1" -> (known after apply)
  # (3 unchanged attributes hidden)
}

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

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.
```

14. Ejecutamos el siguiente comando para aplicar los cambios

- terraform apply

```
jonatan@ubuntu:~/practica terraform/practica 01$ terraform apply
local_file.cali: Refreshing state... [id=eefe216098c4bffe9b9ca742d6bc6234633cb7a1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
/- destroy and then create replacement

Terraform will perform the following actions:

# local_file.cali must be replaced
/- resource "local_file" "cali" {
  ~ content      = "Hola, grupo devops" -> "Hola, grupo devops 2024, este es un ejemplo de terraform" # forces replacement
  ~ content_base64sha256 = "/PnI3wZGKLGeBva8zuYmVpooGKP8fJ35lyLVFDMx0oE=" -> (known after apply)
  ~ content_base64sha512 = "r3paWlUPLmLfkIz95vkkrbXV7Znoac2WpmiGGYzHFXj4ty46SR15Q0HkQLMaj4c6cqPT6GfdbFmJThRRBRw==" -> (known after apply)
  ~ content_md5      = "b22d347a9da1f833729031c1ef76f577" -> (known after apply)
  ~ content_sh1      = "eefe216098c4bffe9b9ca742d6bc6234633cb7a1" -> (known after apply)
  ~ content_sh256     = "fc9c89b75b318a2c67bcbda873b989d5a8a8e28fd1f2529722ef1433313a81" -> (known after apply)
  ~ content_sha512    = "af7a5a588b8f2e69459258b3f52be492b6d757b667a1a7365a99a21866331c5e3e2dc8be924754903879102d66a3e1ce9ca8f4fa19f75b14c6e34e1f11441af" -> (known after apply)
  ~ id              = "eefe216098c4bffe9b9ca742d6bc6234633cb7a1" -> (known after apply)
  # (3 unchanged attributes hidden)
}

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

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

local_file.cali: Destroying... [id=eefe216098c4bffe9b9ca742d6bc6234633cb7a1]
local_file.cali: Destruction complete after 0s
local_file.cali: Creating...
local_file.cali: Creation complete after 0s [id=143d06c08d21c48f4fa913d424c9dd0971b58b]

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

Este documento fue elaborado por SETI para el cliente TCC. Prohibida su reproducción total o parcial sin previa autorización del autor.



15. Ejecutamos el siguiente comando para validar si hay cambios pendientes por aplicar.

- terraform plan
- Como vemos no hay cambios pendientes.

```
jonatan@ubuntu:~/practica_terraform/practica_01$ terraform plan
local_file.cali: Refreshing state... [id=143d06c08d231c48fb4fa913d424c9dd0971b58b]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
jonatan@ubuntu:~/practica_terraform/practica_01$
```

16. Ejecutamos el siguiente comando para ver los cambios en el archivo.

- cat seti.txt

```
jonatan@ubuntu:~/practica_terraform/practica_01$ cat seti.txt
Hola, grupo devops 2024, este es un ejemplo de terraformjonatan@ubuntu:~/practica_terraform/practica_01$
```

17. Ahora queremos eliminar la infraestructura del código

- terraform destroy

```
jonatan@ubuntu:~/practica_terraform/practica_01$ terraform destroy
local_file.cali: Refreshing state... [id=143d06c08d231c48fb4fa913d424c9dd0971b58b]

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:

# local_file.cali will be destroyed
resource "local_file" "cali" {
  content      = "Hola, grupo devops 2024, este es un ejemplo de terraform" -> null
  content_base64sha256 = "VpN9HoBSIA5/eqlVZferpVX0zMcMOUR2TAZYUfnTek=" -> null
  content_base64sha512 = "wG1ahLC63PyogaV6pRj2UfQ383VYN/8Fa4AKXoBRlpIAKHh113aso1V+0smFRX22llg20aUPSSoJ+wdSIDxA==" -> null
  content_md5      = "b2bb15cfe2445c812ffcbbfac869f6a9" -> null
  content_shal     = "143d06c08d231c48fb4fa913d424c9dd0971b58b" -> null
  content_sha256   = "56937d1e8052200b377aa2ef65f11a6f5cecc731c583511d930196147e64de9" -> null
  content_sha512   = "c06d5a8650badcfcaa81a57aa518f651f402dfcdd560dffc15ae002835e8051d692002878759776aca2257ed2c985457db6965836d1a50f492a09fb0752203c4" -> null
  directory_permission = "0777" -> null
  file_permission    = "0777" -> null
  filename          = "seti.txt" -> null
  id                = "143d06c08d231c48fb4fa913d424c9dd0971b58b" -> null
}

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

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

local_file.cali: Destroying... [id=143d06c08d231c48fb4fa913d424c9dd0971b58b]
local_file.cali: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```





18. Ejecutamos el siguiente comando para ver los archivos del directorio

- `ls -ltr`
- Como vemos se generó un archivo "terraform.tfstate.backup" que ya más adelante veremos su función.

```
jonatan@ubuntu:~/practica_terraform/practica_01$ ls -ltr
total 12
-rw-rw-r-- 1 jonatan jonatan 142 feb 22 09:30 terraform.tf
-rw-rw-r-- 1 jonatan jonatan 1472 feb 22 10:21 terraform.tfstate.backup
-rw-rw-r-- 1 jonatan jonatan 180 feb 22 10:21 terraform.tfstate
jonatan@ubuntu:~/practica_terraform/practica_01$
```