



**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 2:.....</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 2:

En este ejercicio usaremos el Provider "random" con el Provider del ejercicio anterior "local\_file"

1. Ejecutamos el siguiente comando para hacer una copia del directorio "practica\_01"

- `cp -a practica_01 practica_02`
- `ls -l`

```
jonatan@ubuntu:~/practica_terraform$ cp -a practica_01 practica_02
jonatan@ubuntu:~/practica_terraform$ ls -l
total 8
drwxrwxr-x 3 jonatan jonatan 4096 feb 22 10:21 practica_01
drwxrwxr-x 3 jonatan jonatan 4096 feb 22 10:21 practica_02
jonatan@ubuntu:~/practica_terraform$
```

2. Ejecutamos los siguientes comandos para ingresar en el directorio "practica\_02", mirar su contenido e ir a Visual Studio Code

- `cd practica_02`
- `ls -l`
- `code .`

```
jonatan@ubuntu:~/practica_terraform$ cd practica_02/
jonatan@ubuntu:~/practica_terraform/practica_02$ ls
terraform.tf  terraform.tfstate  terraform.tfstate.backup
jonatan@ubuntu:~/practica_terraform/practica_02$ code .
jonatan@ubuntu:~/practica_terraform/practica_02$
```

3. Estando en Visual Studio Code, vamos al archivo terraform.tf y hacemos lo siguiente

- Copiamos el siguiente código tal cual nos quede así.

```
terraform.tf x
terraform.tf > resource "random_string" "sufijo"
1  resource "random_string" "sufijo" {
2
3      length = 4
4      special = false
5      upper = false
6      numeric = false
7  }
8
9
10 resource "local_file" "cali" {
11     content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
12     filename = "seti-${random_string.sufijo.id}.txt"
13 }
14
```



- El primer resource usamos el provider "random\_string", esto nos crea un string random de longitud 4, sin caracteres especiales, ni numéricos.
  - En el segundo resource usamos el provider "local\_file", esto nos crea un archivo .txt, el cual tiene por nombre "seti-\${random\_string.sufijo.id}", el cual hace el llamado al provider random\_string, llama el nombre del recurso que en este caso es "sufijo", y seguido a esto llama a "id" que es el que tiene el valor del primer resource.
4. Vamos a la consola y ejecutamos los siguientes comandos estando en el directorio "ejercicio\_02"
- terraform init, para la descarga de los recursos del nuevo provider a usar.
  - terraform plan, para cargar los cambios.
  - terraform apply, para aplicar los cambios.
  - ls -ltr, para ver los archivos del directorio.

```
jonatan@ubuntu:~/practica_terraform/practica_02$ ls -ltr
total 16
-rw-rw-r-- 1 jonatan jonatan 181 feb 22 16:03 terraform.tfstate.backup
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 16:03 seti-krpk.txt
-rw-rw-r-- 1 jonatan jonatan 2284 feb 22 16:03 terraform.tfstate
-rw-rw-r-- 1 jonatan jonatan 295 feb 22 16:09 terraform.tf
jonatan@ubuntu:~/practica_terraform/practica_02$
```

- Como vemos el archivo "seti.txt" ahora se llama "seti-krpk.txt" debido a la configuración que hicimos anteriormente.
- Ahora procedemos a escribir el comando cat.

```
jonatan@ubuntu:~/practica_terraform/practica_02$ cat seti-krpk.txt
Hola, grupo devops 2024, este es un ejemplo de terraformjonatan@ubuntu:~/practica_terraform/practica_02$
```

- Ahora queremos cambiar el sufijo del nombre del archivo seti para ello escribimos.
- ls -ltr, vemos el sufijo del archivo .txt, en este caso el sufijo es "krpk"

```
jonatan@ubuntu:~/practica_terraform/practica_02$ ls -ltr
total 16
-rw-rw-r-- 1 jonatan jonatan 181 feb 22 16:03 terraform.tfstate.backup
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 16:03 seti-krpk.txt
-rw-rw-r-- 1 jonatan jonatan 2284 feb 22 16:03 terraform.tfstate
-rw-rw-r-- 1 jonatan jonatan 295 feb 22 16:09 terraform.tf
```



- Escribimos el comando terraform destroy
- Escribimos el comando terraform apply
- ls -ltr

```
jonatan@ubuntu:~/practica_terraform/practica_02$ ls -ltr
total 16
-rw-rw-r-- 1 jonatan jonatan 295 feb 22 16:09 terraform.tf
-rw-rw-r-- 1 jonatan jonatan 181 feb 22 21:00 terraform.tfstate.backup
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 21:00 seti-fqtd.txt
-rw-rw-r-- 1 jonatan jonatan 2284 feb 22 21:00 terraform.tfstate
jonatan@ubuntu:~/practica_terraform/practica_02$
```

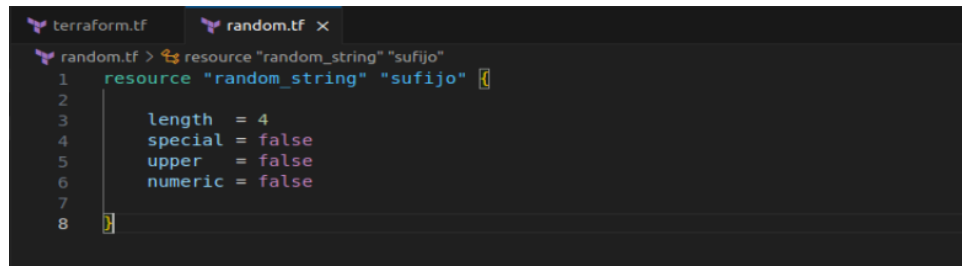
- Ahora el sufijo es "fqtd".
5. Ejecutamos el siguiente comando nos permite mirar el contenido del estado del terraform previo, si no tenemos nunca dato cargado, no mostrara nada, en este caso nos muestra los datos que hay en el archivo "terraform.tf"
- terraform show

```
jonatan@ubuntu:~/practica_terraform/practica_02$ terraform show
# local file.cali:
resource "local_file" "cali" {
  content          = "Hola, grupo devops 2024, este es un ejemplo de terraform"
  content_base64sha256 = "VpW0H0B8S1As/eqlvZfERpVX0zHMcW0UR2TAZYUfmTek="
  content_base64sha512 = "wG1ahLC63PyqgaV6pRj2Uf0C383VYN/8Fa4AKDXoBR1pIAK9h1l3asoiv+0smFRX22llg20aUPSSoJ+wdSIDxA=="
  content_md5        = "b2bb15cfe2445c812ffcfac860fe09"
  content_sh1         = "143d06c08d231c48fb4fa913d424c9dd0971b58b"
  content_sha256      = "56937d1e8052200b3f7aa2ef65f111a6f5cecc731c583511d930196147e64de9"
  content_sha512      = "c06d5a8650badcfcaa81a57aa518f651f402dfcdd560dfc15ae002835e8051d692002878759776aca2257ed2c985457db6965836d1a50f492a09fb0752203c4"
  directory_permission = "0777"
  file_permission      = "0777"
  filename            = "seti-fqtd.txt"
  id                   = "143d06c08d231c48fb4fa913d424c9dd0971b58b"
}

# random_string.sufijo:
resource "random_string" "sufijo" {
  id          = "fqtd"
  length      = 4
  lower       = true
  min_lower   = 0
  min_numeric = 0
  min_special = 0
  min_upper   = 0
  number      = false
  numeric     = false
  result      = "fqtd"
  special     = false
  upper       = false
}
jonatan@ubuntu:~/practica_terraform/practica_02$
```

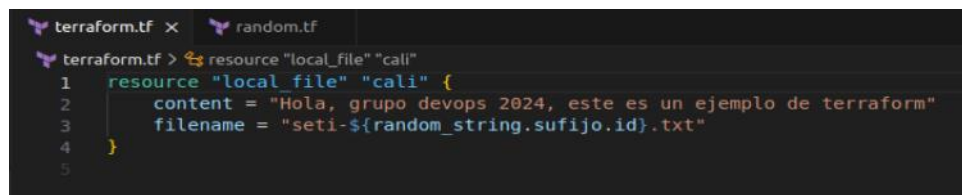


6. Ahora vamos a dividir en contenido de archivo "terraform.tf"
- Creamos un archivo "random.tf" en el vamos a copiar la siguiente parte del código de archivo "terraform.tf", el archivo creado nos debe quedar de la siguiente manera.



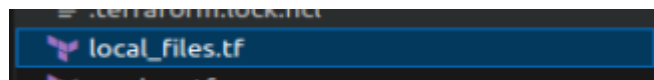
```
random.tf > resource "random_string" "sufijo"
1 resource "random_string" "sufijo" {
2
3     length = 4
4     special = false
5     upper = false
6     numeric = false
7
8 }
```

7. Volvemos al archivo "terraform.tf", este archivo nos debe quedar de la siguiente manera.



```
terraform.tf > resource "local_file" "cali"
1 resource "local_file" "cali" {
2     content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
3     filename = "seti-${random_string.sufijo.id}.txt"
4 }
5
```

8. Ahora vamos a cambiar el nombre del archivo "terraform.tf"
- local\_files.tf



9. Ahora vamos a cambiar el nombre del archivo "terraform.tf"
- Local\_files.tf

10. Ahora vamos a consola, escribimos lo siguiente
- terraform apply
  - Como se podrá ver terraform no nota cambios debido a que es indiferente para terraform tener 1 solo archivo o tener 1000 archivos, lo recomendable es no tener todo en uno, pero tampoco tener 1000 archivos, tener archivos balanceados por recursos, por su ambiente, etc...





```
jonatan@ubuntu:~/practica_terraform/practica_02$ terraform apply
random_string.sufijo: Refreshing state... [id=fqtd]
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.

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

11. Ahora veremos el siguiente comando de terraform y nos daremos de cuenta que todo sigue igual.

- terraform show

```
jonatan@ubuntu:~/practica_terraform/practica_02$ terraform show
# local_file.cali:
resource "local_file" "cali" {
  content      = "Hola, grupo devops 2024, este es un ejemplo de terraform"
  content_base64sha256 = "VpM9H0B5IAs/eqlVzFERpVX0zHMcDUR2TAZYUfmTek="
  content_base64sha512 = "wG1ahLC63PyggaV6pRj2Uf0C383VYN/8Fa4AKDXoBR1pIAKHh1l3asoIV+0smFRX22llg20aUPSSoJ+wdSIDxA=="
  content_md5      = "b2bb15cfe2445c812ffcbbfac869f6a9"
  content_sha1     = "143d06c08d231c48fb4fa913d424c9dd0971b58b"
  content_sha256   = "56937d1e0852200b3f7aa2ef65f11a6f5cecc731c583511d930196147e64de9"
  content_sha512   = "c06d5a8650badcfcaa81a57aa518f651f402dfcdd560dfc15ae002835e0051d692002878759776aca225ed2c985457db6965836d1a50f492a09fb0752203c4"
  directory_permission = "0777"
  file_permission   = "0777"
  filename         = "seti-fqtd.txt"
  id               = "143d06c08d231c48fb4fa913d424c9dd0971b58b"
}

# random_string.sufijo:
resource "random_string" "sufijo" {
  id          = "fqtd"
  length     = 4
  lower      = true
  min_lower  = 0
  min_numeric = 0
  min_special = 0
  min_upper  = 0
  number     = false
  numeric    = false
  result     = "fqtd"
  special    = false
  upper     = false
}
jonatan@ubuntu:~/practica_terraform/practica_02$
```