



Laboratorio
PRACTICA
TERRAFORM



CONTROL DE VERSIONES

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

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

INTRODUCCION	4
PRERREQUISITOS	4
EJERCICIO 2.1:.....	5



INTRODUCCION

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

PRERREQUISITOS

- Tener entendimiento de los temas vistos anteriormente.



EJERCICIO 2.1:

En este ejercicio vamos a crear 5 veces el archivo .txt con sus sufijos diferentes.

1. Ejecutamos el comando, `cd ..`
2. `cp -a practica_02 practica_02.1`
3. `cd practica_02.1`
4. Escribimos el comando "code ."
1. Volvemos a la consola
2. `terraform init`
3. `terraform plan`
4. `terraform apply`
5. `terraform show`
6. `ls -ltr`

```
jonatan@ubuntu:~/practica_terraform/practica_02.1$ ls -ltr
total 44
-rw-rw-r-- 1 jonatan jonatan 619 feb 22 22:34 random.tf
-rw-rw-r-- 1 jonatan jonatan 833 feb 22 22:36 local_files.tf
-rw-rw-r-- 1 jonatan jonatan 2284 feb 22 22:37 terraform.tfstate.backup
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 22:37 seti-ikzk.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 22:37 seti-byiy.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 22:37 seti-ouym.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 22:37 seti-ajju.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 22:37 seti-ddlu.txt
-rw-rw-r-- 1 jonatan jonatan 10718 feb 22 22:37 terraform.tfstate
jonatan@ubuntu:~/practica_terraform/practica_02.1$
```

7. Como vimos se crearon los 5 archivos con su sufijo para cada uno diferente.
8. Ahora vamos al código, y vemos la forma en que está planteado el código y vemos que no es para nada el uso de las buenas prácticas.
9. Para arreglar esto, veremos **DRY: Don't Repeat Yourself**
 - El principio DRY es un concepto fundamental en la programación que enfatiza la importancia de la reutilización del código.
10. Vamos a "local_files.tf", y como vemos esta así:

```
random.tf  local_files.tf x
local_files.tf
1  resource "local_file" "cali-1" {
2    content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
3    filename = "seti-${random_string.sufijo-1.id}.txt"
4  }
5  resource "local_file" "cali-2" {
6    content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
7    filename = "seti-${random_string.sufijo-2.id}.txt"
8  }
9  resource "local_file" "cali-3" {
10   content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
11   filename = "seti-${random_string.sufijo-3.id}.txt"
12 }
13 resource "local_file" "cali-4" {
14   content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
15   filename = "seti-${random_string.sufijo-4.id}.txt"
16 }
17 resource "local_file" "cali-5" {
18   content = "Hola, grupo devops 2024, este es un ejemplo de terraform"
19   filename = "seti-${random_string.sufijo-5.id}.txt"
20 }
```

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



11. Lo dejaremos así:

```
random.tf  local_files.tf x
local_files.tf
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 }
```

12. Vamos a "random.tf", y como vemos esta así:

```
random.tf x  local_files.tf
random.tf
1 resource "random_string" "sufijo-1" {
2
3   length = 4
4   special = false
5   upper = false
6   numeric = false
7
8 }
9 resource "random_string" "sufijo-2" {
10
11   length = 4
12   special = false
13   upper = false
14   numeric = false
15
16 }
17 resource "random_string" "sufijo-3" {
18
19   length = 4
20   special = false
21   upper = false
22   numeric = false
23
24 }
25 resource "random_string" "sufijo-4" {
26
27   length = 4
28   special = false
29   upper = false
30   numeric = false
31
32 }
```



13.Lo dejaremos así:

```
random.tf × local_files.tf
random.tf
1 resource "random_string" "sufijo" {
2
3     length = 4
4     special = false
5     upper = false
6     numeric = false
7 }
8
9
```

14.Ahora, en el archivo "random.tf", agregaremos lo siguiente dentro del resource:

- count = 5

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

15.Ahora, en el archivo "local_files.tf", agregaremos lo siguiente dentro del resource:

- count = 5

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



16. Ahora, en el mismo archivo agregaremos lo siguiente dentro del resource:
- [count.index]

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

17. Ahora vamos a consola, y escribimos los siguiente:
- terraform apply
 - ls -ltr

```
jonatan@ubuntu:~/practica_terraform/practica_02.1$ ls -ltr
total 52
-rw-rw-r-- 1 jonatan jonatan 134 feb 22 23:43 random.tf
-rw-rw-r-- 1 jonatan jonatan 190 feb 22 23:50 local_files.tf
-rw-rw-r-- 1 jonatan jonatan 10718 feb 22 23:50 terraform.tfstate.backup
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 23:50 seti-yhao.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 23:50 seti-qlvt.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 23:50 seti-affy.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 23:50 seti-cwks.txt
-rwxrwxr-x 1 jonatan jonatan 56 feb 22 23:50 seti-lrky.txt
-rw-rw-r-- 1 jonatan jonatan 9412 feb 22 23:50 terraform.tfstate
jonatan@ubuntu:~/practica_terraform/practica_02.1$
```

18. En un ejemplo, en vez de 5, sean 50, en vez de ponernos a copiar 50 veces lo mismo, de esta manera en la que actualizamos el proceso se genera una mejor practica optimizando código.
19. Escribimos el comando "terraform show", veremos la posición de los índices de cada uno.



```
# random_string.sufijo[0]:
resource "random_string" "sufijo" {
  id          = "lrqy"
  length      = 4
  lower       = true
  min_lower   = 0
  min_numeric = 0
  min_special = 0
  min_upper   = 0
  number      = false
  numeric     = false
  result      = "lrqy"
  special     = false
  upper       = false
}

# random_string.sufijo[1]:
resource "random_string" "sufijo" {
  id          = "affy"
  length      = 4
  lower       = true
  min_lower   = 0
  min_numeric = 0
  min_special = 0
  min_upper   = 0
  number      = false
  numeric     = false
  result      = "affy"
  special     = false
  upper       = false
}

# random_string.sufijo[2]:
resource "random_string" "sufijo" {
  id          = "cwks"
  length      = 4
  lower       = true
  min_lower   = 0
  min_numeric = 0
  min_special = 0
  min_upper   = 0
  number      = false
  numeric     = false
  result      = "cwks"
}
```

Nota: Ya en este punto hemos visto 2 tipos de Providers, el principio DRY, sintaxis de terraform, manejo de comandos.