

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 9:	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 9:

En este ejercicio vamos a usar Target Resources.

En Terraform, el concepto de "Target Resources" se refiere a la capacidad de limitar las operaciones de Terraform a un conjunto específico de recursos dentro de tu configuración. Esto es útil cuando deseas realizar operaciones como la creación, actualización o destrucción solo en un subconjunto de recursos en lugar de todos los recursos definidos en tu configuración.

Hacemos una copia del ejercicio anterior, ya agregamos lo siguiente en el archivo "vpc.tf", línea 24 a la 26 (depends_on esta explicado, Tema: **Dependencias**, en el archivo "teoria.pdf").

```
resource "aws_subnet" "private_subnet" {

vpc_id = aws_vpc.vpc_virginia.id

cidr_block = var.subnets[1]

tags = {

Name = "Private subnet"

}

depends_on = [

aws_subnet.public_subnet

]

27
```

- 1. Ejecutamos terraform plan
- 2. Ejecutamos terraform apply --auto-approve=true
 - Esto nos permite omitir el proceso de aprobación, no es recomendado hacerlo.
- 3. Ingresamos a la página de AWS, con el usuario IAM y veremos la creación de las dos VPC y las dos subredes.



4. Ahora nos dirigimos al archivo "vpc.tf" y editamos los nombres de las subredes agregándole un temp al final:

5. Volvemos a la consola y ejecutamos "terraform plan" y veremos que nos marca los 2 cambios hechos.

```
Terraform will perform the following actions:
   aws_subnet.private_subnet will be updated in-place
   resource "aws_subnet" "private_subnet" [
                                                      = "subnet-090d230d27934793e"
       id
       tags "Name" = "Private subnet" -- "Private subnet temp"
       tags_all
                         = "Private subnet" -> "Private subnet temp"
 # aws_subnet.public_subnet will be updated in-place
             "aws_subnet"
                         "public_subnet" {
                                                      = "subnet-86c94b9ad2bac5d94"
       tags
"Name" = "Public subnet" -- "Public subnet temp"
       tags_all
                        = "Public subnet" -- "Public subnet temp"
lan: 0 to add, 2 to change, 0 to destroy.
```



- 6. Ahora vamos a ejecutar un solo cambio, No los 2, para ello ejecutamos el siguiente comando:
 - "terraform apply --target aws.subnet.public_subnet".
 - Nos indica una advertencia de solo aplicamos un cambio y no aplicamos todos los cambios.

```
Place B to add, 1 to charge, 8 to destroy.

Northy: Resource targeting is in effect

You are creating a plan with the 'target option, which means that the result of this plan may not represent all of the charges requested by the current configuration.

The 'target option is not for routine use, and is provided only for exceptional situations such as recovering from errors or mistakes, or when Terraforn specifically suggests to use it as part of an error measage.

Do you want to perform these actions?
Ferraforn will perform these actions?
Ferraforn will perform these actions described above.

Dely 'yes' will be accepted to approve.

Enter a value: yes

way submet.public_submet: Andifysiag... [id=submet-0cc9409ad2Mac5d94]

way.submet.public_submet: Andifysiag... [id=submet-0cc9409ad2Mac5d94]

Warning: Applied changes may be incomplete

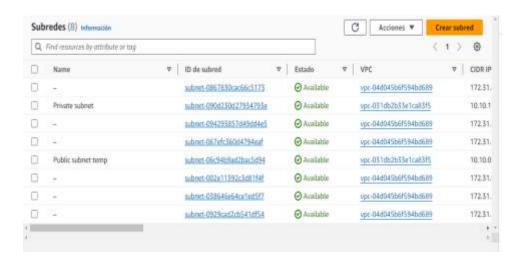
The plan was created with the 'target option in effect, so some charges requested in the configuration may have been ignored and the output values may not be fully upfated. Non the following command to verify that no other changes are pending:

terraform plan

Note that the 'target option is not suitable for routine use, and is provided only for exceptional situations such as recovering from errors or mislakes, or when Terraform specifically suggests to use it as part of an error message.

Note: The that the 'target option is not suitable for routine use, and is provided only for exceptional situations such as recovering from errors or mislakes, or when Terraform specifically suggests to use it as part of an error message.
```

7. Ahora vamos a la página y verificamos las subredes y veremos el cambio.





8. En consola si ejecutamos un terraform plan, veremos que tenemos un cambio todavía pendiente.

```
jonatempubustu:-_fundomentatio_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_personation_person
```

9. Opcional, Nos dirigimos al archivo "vpc.tf" y dejamos los nombres como los teníamos al principio.

```
resource "aws_subnet" "public_subnet" {
    vpc_id = aws_vpc.vpc_virginia.id
    cidr_block = var.subnets[0]

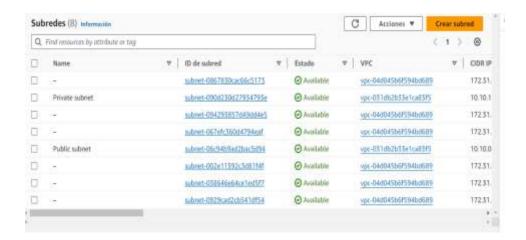
# para asiganr ip publica: map_public_ip_on_launch
# por defecto siempre esta en false
map_public_ip_on_launch = true
tags = {
    Name = "Public subnet"
}

resource "aws_subnet" "private_subnet" {
    vpc_id = aws_vpc.vpc_virginia.id
    cidr_block = var.subnets[1]
    tags = {
        Name = "Private subnet"
    }

depends_on = [
        aws_subnet.public_subnet
]
```



10.En consola ejecutamos "terraform plan", después ejecutamos "terraform apply", ingresamos a las subredes y verificamos que los nombres volvieron a como estaban al principio.



11. Por ultimo ejecutamos el comando terraform destroy.