

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 7:	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 7:

Ejecutamos los siguientes comandos en consola.

```
jonatan@ubuntu:~$ cd practica_terraform/
jonatan@ubuntu:~/practica_terraform$ cp -a practica_06 practica_07
jonatan@ubuntu:~/practica_terraform$ cd practica_07
jonatan@ubuntu:~/practica_terraform/practica_07$ code .
jonatan@ubuntu:~/practica_terraform/practica_07$
```

1. Borramos las siguientes líneas del archivo "providers.tf"

2. Borramos las siguientes líneas del archivo "vpc.tf".

```
vpc.tf > % resource "aws_vpc" "vpc_ohio"

1    resource "aws_vpc" "vpc_virginia" {
2       cidr_block = var.virginia_cidr
3       tags = {
4          Name = "VPC_VIRGINIA"
5          env = "Dev"
6       }
7    }
8
9    resource "aws_vpc" "vpc_ohio" {
10       cidr_block = var.ohio_cidr
11       tags = {
12          Name = "VPC_OHIO"
13          env = "Dev"
14       }
15       provider = aws.Ohio
16    }
```



3. Borramos la siguiente línea del archivo "variables.tf".

4. Borramos la siguiente línea del archivo "terraform.tfvars".

```
providers.tf × vpc.tf variables.tf terraform.tfvars 1 •

terraform.tfvars > m ohio_cidr

virginia_cidr = "10.10.0.0/16"

ohio_cidr = "10.20.0.0/16"
```

5. Modificamos el archivo "variables.tf", para que nos quede de la siguiente manera.

```
providers.tf x vpc.tf variables.tf x

variables.tf > % variable "virginia_cidr" > 69 sensitive

1 variable "virginia_cidr" {
2 description = "CIDR Virginia"
3 type = string
4 sensitive = false
5
```

6. Nos dirigimos a la consola y ejecutamos "terraform plan".



7. Nos dirigimos al archivo "variables.tf", cambiamos el valor de sensitive a true.

```
providers.tf vpc.tf variables.tf x

variables.tf > ...

variable "virginia cidr" {
 description = "CIDR Virginia"
 type = string
 sensitive = true
}
```

8. Nos dirigimos a la consola y ejecutamos "terraform plan", y miramos que nos muestra "sensitive value".

9. Nos dirigimos al archivo "vpc.tf" y agregamos lo siguiente.



10. Nos dirigimos al archivo "variables.tf" y agregamos lo siguiente.

```
providers.tf
                w vpc.tf
                                y variables.tf ×
🦖 variables.tf > 😭 variable "private_subnet" > 🖭 description
      variable "virginia cidr" {
           description = "CIDR Virginia"
           type = string
           sensitive = true
      variable "public subnet" {
         description = "CIDR public subnet"
         type = string
      variable "private subnet" [
 11
         description = "CIDR private subnet"
 12
 13
         type = string
```

11. Nos dirigimos al archivo "terraform.tfvars" y agregamos lo siguiente.

```
providers.tf vpc.tf variables.tf terraform.tfvars x

terraform.tfvars >  private_subnet
    virginia_cidr = "10.10.0.0/16"
    public_subnet = "10.10.0.0/24"
    private_subnet = "10.10.1.0/24"
```

12. Nos dirigimos al archivo "vpc.tf" y agregamos la siguiente línea.



- 13. En el mismo archivo "vpc.tf", agregamos la siguiente línea (13).
 - comentario se hace con #.

```
resource "aws_subnet" "public_subnet" {
    vpc_id = aws_vpc.vpc_virginia.id
    cidr_block = var.public_subnet
    # para asiganr ip publica: map_public ip on launch
    # por defecto siempre esta en false
    map_public_ip_on_launch = true
}
```

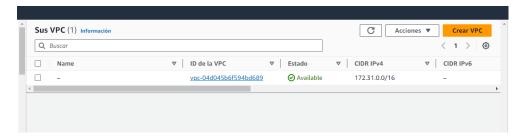
14. En el mismo archivo "vpc.tf", agregamos un nuevo resourse.

```
resource "aws_subnet" "private_subnet" {
    vpc_id = aws_vpc.vpc_virginia.id
    cidr_block = var.private_subnet
}
```

- 15. Ejecutamos en consola, terraform plan.
 - Como veremos nos crea 3, la privada, publica y la vpc de virginia.



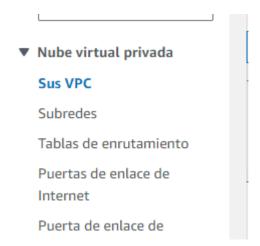
16. Entramos en la página de Aws con el usuario iam, a nuestras vpc y veremos que solo tenemos la predeterminada.



- 17. Ejecutamos en consola, terraform apply, damos la confirmación "yes".
- 18. Nos dirigimos de nuevo a la página de aws, actualizamos y veremos la VPC de Virginia.

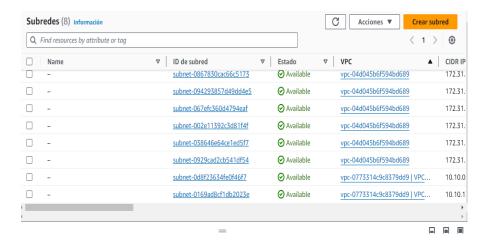


19. Ingresamos al menú de VPC de subnets.





20. Damos click en VPC con la flecha arriba, y veremos las 2 subnets creadas de ultimas.



21. Vamos al archivo "variables.tf" y vamos a empezar a usar list, para ello vamos a editar el archivo.

```
variables.tf > variable "subnets" > variables.tf x

variable "virginia_cidr" {
    description = "CIDR Virginia"
    type = string
    sensitive = true
}

# variable "public_subnet" {
    # description = "CIDR public subnet"
# type = string
# type =
```



22. Vamos al archivo "terraform.tfvars" y vamos a editar el archivo para que quede de la siguiente forma.

```
providers.tf x vpc.tf variables.tf terraform.tfvars x

terraform.tfvars > [ ] subnets

virginia_cidr = "10.10.0.0/16"

public_subnet = "10.10.0.0/24"

private_subnet = "10.10.1.0/24"

subnets = [] "10.10.0.0/24", "10.10.1.0/24"]
```

23. Vamos al archivo "vpc.tf y vamos a editar el archivo en las líneas 13 y 18.

```
8  resource "aws_subnet" "public_subnet" {
9     vpc_id = aws_vpc.vpc_virginia.id
10     cidr_block = var.subnets[0]
11     # para asiganr ip publica: map_public_ip_on_launch
12     # por defecto siempre esta en false
13     map_public_ip_on_launch = true
14  }
15
16  resource "aws_subnet" "private_subnet" {
17     vpc_id = aws_vpc.vpc_virginia.id
18     cidr_block = var.subnets[1]
19 }
```

24. Nos dirigimos a la consola y ejecutamos terraform plan, como veremos nos dice que no hay cambios realizados, lo único que hicimos fue acomodar el código y expresarlo de otra forma.

```
jonatan@ubuntu:~/practica_terraform/practica_07$ terraform plan
aws_vpc.vpc_virginia: Refreshing state... [id=vpc-0773314c9c8379dd9]
aws_subnet.private_subnet: Refreshing state... [id=subnet-0169ad8cf1db2023e]
aws_subnet.public_subnet: Refreshing state... [id=subnet-0d8f23634fe0f46f7]

No changes. Your infrastructure matches the configuration.

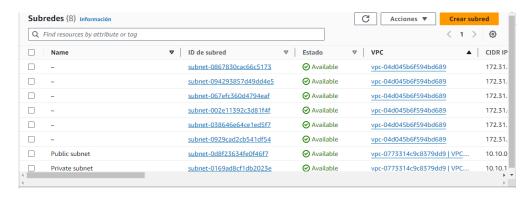
Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
jonatan@ubuntu:~/practica_terraform/practica_07$ ■
```



25. Ahora agregaremos tags a la pública y privada.

```
providers.tf
                               yariables.tf
                                                terraform.tfvars
                ypc.tf
🦖 vpc.tf > 😭 resource "aws_subnet" "private_subnet" > 긂 tags > 🖭 env
      resource "aws vpc" "vpc virginia" {
         cidr block = var.virginia cidr
         tags = {
           Name = "VPC VIRGINIA"
           env = "Dev"
      resource "aws subnet" "public subnet" {
        vpc id = aws vpc.vpc virginia.id
        cidr_block = var.subnets[0]
        map public ip on launch = true
         tags = {
           Name = "Public subnet"
           env = "Dev"
                                "private_subnet" (name)
                               Reference Name
         vpc id = aws vpc.vpc virginia.id
         cidr block = var.subnets[1]
         tags = {
          Name = "Priavate subnet"
           env = "Dev"
 25
```

- 26. Volvemos a la consola y ejecutamos, terraform plan, terraform apply.
- 27. Volvemos a la página de aws, a subnets con los tags agregados previamente.





- 28. Ahora acomodaremos el código con la variable map.
- 29. Vamos al archivo "variables.tf", vamos a agregar lo siguiente.

```
20
21  variable "tags" {
22     description = "Tags del proyecto"
23     type = map(string)
24 }
```

30. Vamos al archivo "terraform.tfvars", vamos a agregar lo siguiente.

31. Vamos al archivo "vpc.tf", vamos a agregar lo siguiente.

```
providers.tf
                      wpc.tf
                                           variables.tf
                                                                  terraform.tfvars
 🦖 vpc.tf > 😘 resource "aws_subnet" "private_subnet" > 긂 tags > 🖭 Name
         resource "aws vpc" "vpc virginia" {
           cidr_block = var.virginia_cidr
            tags = {
               Name = "VPC VIRGINIA"
         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"
```



32. Vamos al archivo "providers.tf", vamos a agregar los tags generales.

```
providers.tf x vpc.tf variables.tf terraform.tfvars

providers.tf > provider "aws"

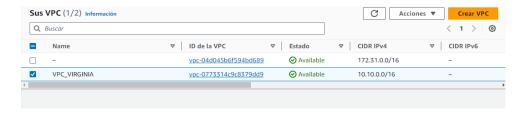
terraform {
    required_providers {
    aws = {
        source = "hashicorp/aws"
        version = ">=5.35.0, <5.37.0, !=5.35.0"
    }

    required_version = "~>1.7.0"

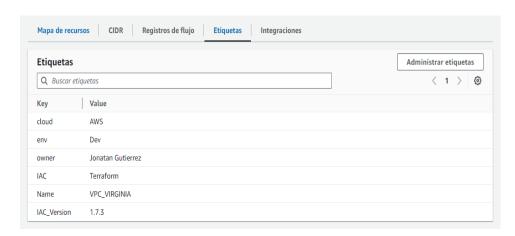
    }

    region = "us-east-1"
    default_tags {
        tags = var.tags
    }
}
```

- 33. Volvemos a la página de AWS, miramos tanto en VPC como en subredes.
- 34. En VPC
 - Damos click en el ID de la VPC



 Ingresamos en etiquetas y veremos los tags de default con el tag de "Name" individual.





- 35. En subredes es el mismo proceso para ver las etiquetas.
- 36. Terraform destroy