

╭─░▒▓ /mnt/c/1/B/1/6. Sesión/3. Post-Work/iGomezP-myAPI-iac  on main !1 ?2 ──────────────────────────────────────────────────────────────────────────────────── ✔  at 04:57:56 PM ▓▒░─╮

╰─ terraform plan -out=tfplan ─╯

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:

# aws\_instance.ec2\_instance will be created

+ resource "aws\_instance" "ec2\_instance" {

+ ami = "ami-03f6d497fceb40069"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ disable\_api\_stop = (known after apply)

+ disable\_api\_termination = (known after apply)

+ ebs\_optimized = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ host\_resource\_group\_arn = (known after apply)

+ id = (known after apply)

+ instance\_initiated\_shutdown\_behavior = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.micro"

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = (known after apply)

+ monitoring = (known after apply)

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ placement\_partition\_number = (known after apply)

+ primary\_network\_interface\_id = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

+ public\_ip = (known after apply)

+ secondary\_private\_ips = (known after apply)

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ tags = {

+ "Environment" = "Develop"

+ "Name" = "dev-server-1"

+ "Owner" = "iGomezP"

}

+ tags\_all = {

+ "Environment" = "Develop"

+ "Name" = "dev-server-1"

+ "Owner" = "iGomezP"

}

+ tenancy = (known after apply)

+ user\_data = (known after apply)

+ user\_data\_base64 = (known after apply)

+ user\_data\_replace\_on\_change = false

+ vpc\_security\_group\_ids = (known after apply)

+ capacity\_reservation\_specification {

+ capacity\_reservation\_preference = (known after apply)

+ capacity\_reservation\_target {

+ capacity\_reservation\_id = (known after apply)

+ capacity\_reservation\_resource\_group\_arn = (known after apply)

}

}

+ ebs\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ snapshot\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

+ enclave\_options {

+ enabled = (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

+ maintenance\_options {

+ auto\_recovery = (known after apply)

}

+ metadata\_options {

+ http\_endpoint = (known after apply)

+ http\_put\_response\_hop\_limit = (known after apply)

+ http\_tokens = (known after apply)

+ instance\_metadata\_tags = (known after apply)

}

+ network\_interface {

+ delete\_on\_termination = (known after apply)

+ device\_index = (known after apply)

+ network\_card\_index = (known after apply)

+ network\_interface\_id = (known after apply)

}

+ private\_dns\_name\_options {

+ enable\_resource\_name\_dns\_a\_record = (known after apply)

+ enable\_resource\_name\_dns\_aaaa\_record = (known after apply)

+ hostname\_type = (known after apply)

}

+ root\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

}

# aws\_subnet.dev-subnet-public-1 will be created

+ resource "aws\_subnet" "dev-subnet-public-1" {

+ arn = (known after apply)

+ assign\_ipv6\_address\_on\_creation = false

+ availability\_zone = "us-west-1a"

+ availability\_zone\_id = (known after apply)

+ cidr\_block = "10.0.1.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

+ ipv6\_cidr\_block\_association\_id = (known after apply)

+ ipv6\_native = false

+ map\_public\_ip\_on\_launch = true

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "dev-subnet-public-1"

}

+ tags\_all = {

+ "Name" = "dev-subnet-public-1"

}

+ vpc\_id = (known after apply)

}

# aws\_vpc.dev-vpc will be created

+ resource "aws\_vpc" "dev-vpc" {

+ arn = (known after apply)

+ cidr\_block = "10.0.0.0/16"

+ default\_network\_acl\_id = (known after apply)

+ default\_route\_table\_id = (known after apply)

+ default\_security\_group\_id = (known after apply)

+ dhcp\_options\_id = (known after apply)

+ enable\_classiclink = (known after apply)

+ enable\_classiclink\_dns\_support = (known after apply)

+ enable\_dns\_hostnames = true

+ enable\_dns\_support = true

+ enable\_network\_address\_usage\_metrics = (known after apply)

+ id = (known after apply)

+ instance\_tenancy = "default"

+ ipv6\_association\_id = (known after apply)

+ ipv6\_cidr\_block = (known after apply)

+ ipv6\_cidr\_block\_network\_border\_group = (known after apply)

+ main\_route\_table\_id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "dev-vpc"

}

+ tags\_all = {

+ "Name" = "dev-vpc"

}

}

# docker\_container.microservice will be created

+ resource "docker\_container" "microservice" {

+ attach = false

+ bridge = (known after apply)

+ command = (known after apply)

+ container\_logs = (known after apply)

+ container\_read\_refresh\_timeout\_milliseconds = 15000

+ entrypoint = (known after apply)

+ env = (known after apply)

+ exit\_code = (known after apply)

+ gateway = (known after apply)

+ hostname = (known after apply)

+ id = (known after apply)

+ image = (known after apply)

+ init = (known after apply)

+ ip\_address = (known after apply)

+ ip\_prefix\_length = (known after apply)

+ ipc\_mode = (known after apply)

+ log\_driver = (known after apply)

+ logs = false

+ must\_run = true

+ name = "microservice-demo"

+ network\_data = (known after apply)

+ read\_only = false

+ remove\_volumes = true

+ restart = "no"

+ rm = false

+ runtime = (known after apply)

+ security\_opts = (known after apply)

+ shm\_size = (known after apply)

+ start = true

+ stdin\_open = false

+ stop\_signal = (known after apply)

+ stop\_timeout = (known after apply)

+ tty = false

+ wait = false

+ wait\_timeout = 60

+ healthcheck {

+ interval = (known after apply)

+ retries = (known after apply)

+ start\_period = (known after apply)

+ test = (known after apply)

+ timeout = (known after apply)

}

+ labels {

+ label = (known after apply)

+ value = (known after apply)

}

+ ports {

+ external = 5000

+ internal = 5000

+ ip = "0.0.0.0"

+ protocol = "tcp"

}

}

# docker\_image.microservice will be created

+ resource "docker\_image" "microservice" {

+ id = (known after apply)

+ image\_id = (known after apply)

+ latest = (known after apply)

+ name = "microservice"

+ output = (known after apply)

+ repo\_digest = (known after apply)

+ build {

+ dockerfile = "Dockerfile"

+ path = "../microservice/."

+ remove = true

+ tag = [

+ "microservice:latest",

]

}

}

Plan: 5 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ public\_ip = (known after apply)

╷

│ Warning: Deprecated attribute

│

│ on main.tf line 29, in resource "docker\_container" "microservice":

│ 29: image = docker\_image.microservice.latest

│

│ The attribute "latest" is deprecated. Refer to the provider documentation for details.

│

│ (and one more similar warning elsewhere)

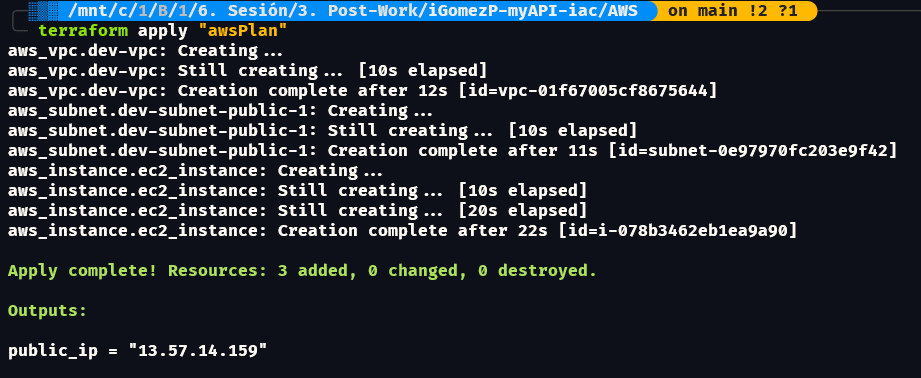
╵

───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:

terraform apply "tfplan"



|  |  |  |
| --- | --- | --- |
| Requisito | Sí lo cumple | No lo cumple |
| a. Tener instalado Terraform en local | ✔️ |  |
| b. Identificar los comandos básicos para el despliegue y destrucción de ambientes | ✔️ |  |
| c. Crear nuestro primer despliegue | ✔️ |  |
| d. Crear un despliegue con un servidor en la nube de tu elección. | ✔️ |  |