

¡Levantamos una infraestructura!

Nuestro objetivo es levantar una infraestructura cloud más robusta, segura y resiliente,
que consista meramente en recursos de infraestructura:

- Un VPC.
- Un Internet Gateway asociado al VPC creado.
- Una subnet pública.
- Una subnet privada.
- Una tabla de ruteo dedicada a la subnet pública.
- Una tabla de ruteo dedicada a la subnet privada.
- Las asociaciones de ambas tablas de ruteo con sus respectivas subnets.
- Un NAT Gateway asociado a la subnet privada.
- Una Elastic IP.

Nuestro código será modularizado, es decir, segmentado en tres archivos: uno para variables, otro para seleccionar el proveedor Cloud y el último que se utilizará para levantar nuestra infra.

Armando el ambiente en tres pasos:

1. Declaración de variables.
2. Declaración del provider a utilizar.
3. Creación de la infraestructura base.

PASO A PASO:

1- Creamos archivos .tf

1.1- Main

```
# Propósito: crear infraestructura AWS
# Autor: DH
# Fecha: 30.07.21
# Versión: 1.0
# =====

# =====
# Creamos nuestro VPC
resource "aws_vpc" "Main" {

# usamos el bloque "resource", el "provider element" y una "etiqueta"
  cidr_block = var.main_vpc_cidr

# le pasamos por variable el CIDR block que quiero que use
  instance_tenancy = "default"

  tags = {
    Name = "My_VPC"
  }

}
# =====
# Creamos un Internet Gateway "Y" lo asociamos al VPC que se acaba de crear
resource "aws_internet_gateway" "IGW" { # Internet Gateway

  vpc_id = aws_vpc.Main.id

# vamos a conocer el vpc_id solo cuando el VPC se haya creado
  tags = {
    Name = "IGW"
  }

}
# =====
# Creamos la subnet pública
resource "aws_subnet" "public_subnets" {

# creamos las subnets públicas
  vpc_id = aws_vpc.Main.id
  cidr_block = var.public_subnets

# CIDR block para mis public subnets
  tags = {
    Name = "Public Subnet"
  }

}
# =====
# Creamos la subnet privada

# creamos nuestras private subnets
resource "aws_subnet" "private_subnets" {
```

```

vpc_id = aws_vpc.Main.id
cidr_block = var.private_subnets

# CIDR block para mis subnets privadas
tags = {
    Name = "Private Subnet"
}
}
# =====
# Tabla de ruteo para la subnet pública
resource "aws_route_table" "Public_RT" {

# Creamos nuestro Route Table para la subnet pública
vpc_id = aws_vpc.Main.id
route {
    cidr_block = "0.0.0.0/0"

# Declaramos el tráfico desde la subnet pública llega a Internet desde el Internet Gateway
    gateway_id = aws_internet_gateway.IGW.id
}
tags = {
    Name = "Tabla de Ruteo Pública"
}
}
# =====
# Tabla de ruteo para la subnet privada
resource "aws_route_table" "Private_RT" {

# Creating RT for Private Subnet
vpc_id = aws_vpc.Main.id
route {
    cidr_block = "0.0.0.0/0"

# Tráfico proveniente desde la subnet privadas llegando a Internet vía NAT Gateway
    nat_gateway_id = aws_nat_gateway.NAT_GW.id
}
tags = {
    Name = "Tabla de Ruteo Privada"
}
}
# =====

# Asociación de tabla de ruteo con la subnet pública
resource "aws_route_table_association" "Public_RT_Association" {
    subnet_id = aws_subnet.public_subnets.id
    route_table_id = aws_route_table.Public_RT.id
}
# =====

# Asociación de tabla de ruteo con la subnet privada
resource "aws_route_table_association" "Private_RT_Association" {
    subnet_id = aws_subnet.private_subnets.id
    route_table_id = aws_route_table.Private_RT.id
}
resource "aws_eip" "NAT_EIP" {
    vpc = true
    tags = {
        Name = "NAT con elastic IP"
    }
}

```

```
# =====

# Creación del NAT Gateway usando subnet_id y allocation_id
resource "aws_nat_gateway" "NAT_GW" {
  allocation_id = aws_eip.NAT_EIP.id
  subnet_id = aws_subnet.public_subnets.id
  tags = {
    Name = "NAT Gateway alocada a la subnet pública"
  }
}
```

1.2 Provider

```
# =====

# Propósito:  declaramos que proveedor cloud queremos usar
# Autor:      DH
# Fecha:      30.07.21
# Versión:    1.0
# =====

# =====
# Declaramos el Cloud Provider con el que queremos trabajar

terraform {

  # Le decimos que queremos:
  # a. la versión del binario de terraform mayor o igual a 0.12

  required_version = ">=0.12"
  required_providers {
    aws = {

      # Especificamos desde donde queremos descargar el binario:
      source = "hashicorp/aws"

      # Le decimos que solo permitirá:
      # b. la versión del binario del provider 3.20.0 (con cierta restricción)
      version = "~> 3.20.0"

    }
  }
}

# =====

# =====
# Declaramos la región donde queremos levantar nuestra infra

provider "aws" {
  shared_credentials_file = "~/.aws/credentials"
  region = "us-east-1"
}

# =====
```

1.3 Variables

```
# =====q
# Proposito:  declaramos todas las variables que vamos a usar
# Autor:     DH
# Fecha:     30.07.21
# Version:    1.0
# =====

variable "aws_region_id" {
  description = "la region"
  type        = string
  default     = "us-east-1"
}
variable "main_vpc_cidr" {
  description = "Nuestro Security Group"
  type        = string
  default     = "10.0.0.0/24"
}
variable "public_subnets" {
  description = "subnet con acceso a internet"
  type        = string
  default     = "10.0.0.128/26"
}
variable "private_subnets" {
  description = "subnet sin acceso a internet"
  type        = string
  default     = "10.0.0.192/26"
}
# =====
```

2- Configuro aws con las keys

```
$ aws configure
AWS Access Key ID [None]: AKIAI455UP5YBUKSRUE
AWS Secret Access Key [None]: JMMiH4lVMNXt6eUp4Lj4by+jA8IK8D6fP9stogrE
Default region name [None]: us-west-1
Default output format [None]: json
```

*asignar región que figure en aws
y salida en json

3- ejecutar: Terraform init, plan y apply

```
Tomi PC@DESKTOP-33VOKL2 MINGW64 ~/Desktop/Tomi/Certified Tech Developer/Bimestre 3/Infraestructura II/terraform
$ terraform apply
```

y se empieza a levantar

```
terraform > terraform apply -input=false
70     route_table_id = aws_route_table.Public_RT.id
71   }
72   # =====
73   # Asociación de tabla de ruteo con la subnet privada
74   resource "aws_route_table_association" "Private_RT_Association" {
75     subnet_id = aws_subnet.private_subnets.id
76     route_table_id = aws_route_table.Private_RT.id
77   }
78   resource "aws_eip" "NAT_EIP" {
79     vpc      = true
80     tags = {
81       Name = "NAT con elastic IP"
82     }
83   }
84   # =====
85   # Creación del NAT Gateway usando subnet_id y allocation_id
86   resource "aws_nat_gateway" "NAT_GW" {
87     allocation_id = aws_eip.NAT_EIP.id
88     subnet_id = aws_subnet.public_subnets.id
89     tags = {
90       Name = "NAT Gateway alojada a la subnet pública"
91     }
92   }
93 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
+ tags                                = {
+   "Name" = "My_VPC"
+ }
}

Plan: 10 to add, 0 to change, 0 to destroy.

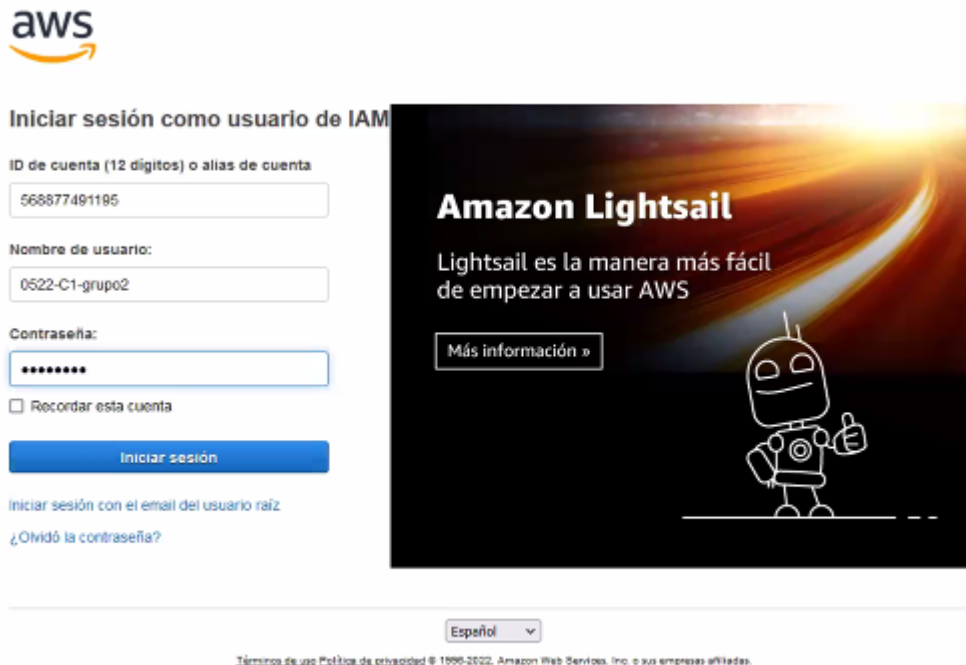
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

aws_eip.NAT_EIP: Creating...
aws_vpc.Main: Creating...
aws_eip.NAT_EIP: Creation complete after 2s [id=eipalloc-075b01cabf93a9287]
```

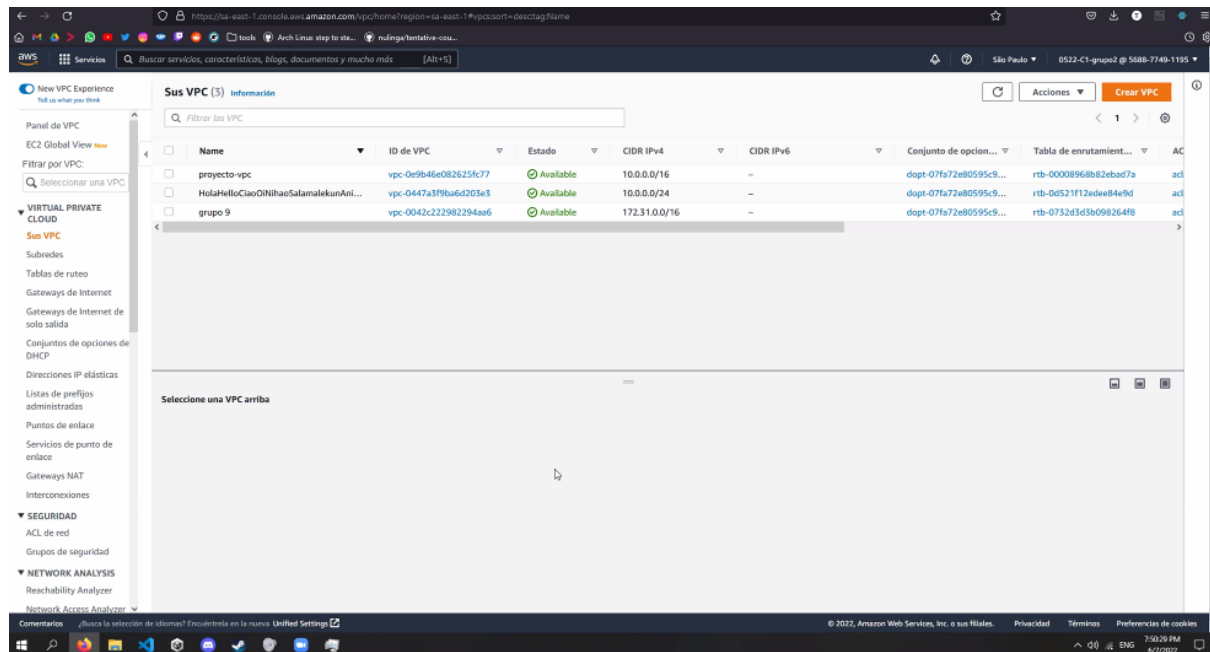
#Verificar que estén creadas las vpc

ingreso en aws



The image shows the AWS login page. On the left, there is a login form with the following fields: 'ID de cuenta (12 dígitos) o alias de cuenta' with the value '568877491195', 'Nombre de usuario:' with the value '0522-C1-grupo2', and 'Contraseña:' with masked characters. There is a checkbox for 'Recordar esta cuenta' and a blue 'Iniciar sesión' button. Below the button, there are links for 'Iniciar sesión con el email del usuario raíz' and '¿Olvidó la contraseña?'. On the right, there is a promotional banner for 'Amazon Lightsail' with the text 'Lightsail es la manera más fácil de empezar a usar AWS' and a 'Más información »' button. The banner also features a cartoon robot character. At the bottom, there is a language selector set to 'Español' and a copyright notice for Amazon Web Services, Inc. © 1996-2022.

verifico vpc (en la región configurada)



The image is a screenshot of the AWS Management Console, specifically the 'Sus VPC (3)' page. The page shows a list of VPCs with columns for Name, ID de VPC, Estado, CIDR IPv4, CIDR IPv6, Conjunto de opciones..., and Tabla de enrutamiento... The table contains three entries:

Nombre	ID de VPC	Estado	CIDR IPv4	CIDR IPv6	Conjunto de opciones...	Tabla de enrutamiento...
proyecto-vpc	vpc-0e9b46e082625fc77	Available	10.0.0.0/16	-	dopt-07fa72e80595c9...	rtb-00008968b82ebad7a
HolaHelloCiaoOINIhaoSalamalekunAni...	vpc-0447a3f9ba6d203e3	Available	10.0.0.0/24	-	dopt-07fa72e80595c9...	rtb-0d521f12edee84e9d
grupo 9	vpc-0042c222982294aa6	Available	172.31.0.0/16	-	dopt-07fa72e80595c9...	rtb-0732a3d3b098264f8

Below the table, there is a section titled 'Seleccione una VPC arriba' with a search bar and a list of VPCs. The page also includes a left-hand navigation menu with options like 'Panel de VPC', 'EC2 Global View', 'Filtrar por VPC', 'VIRTUAL PRIVATE CLOUD', 'Sus VPC', 'Subredes', 'Tablas de ruteo', 'Gateways de Internet', 'Gateways de Internet de solo salida', 'Conjuntos de opciones de DHCP', 'Direcciones IP elásticas', 'Listas de prefijos administradas', 'Puntos de enlace', 'Servicios de punto de enlace', 'Gateways NAT', 'Interconexiones', 'SEGURIDAD', 'ACL de red', 'Grupos de seguridad', 'NETWORK ANALYSIS', and 'Resiliability Analyzer'. The bottom of the page shows the AWS logo, a search bar, and a footer with copyright information for Amazon Web Services, Inc. © 2022.