

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Actividad Abierta Laboratorio: despliegue de MEAN multicapa mediante Terraform

Objetivos: En esta actividad aprenderás a utilizar Terraform con un ejemplo sencillo pero completo. Para ello, usaremos un stack MEAN con dos o más máquinas. A través de esta actividad conseguirás familiarizarte con la herramienta Terraform para desplegar conjuntos de componentes que se comuniquen entre ellos.

Rúbrica: Los criterios de evaluación de la práctica son:

Actividad	Descripción	Puntos	Peso	Cumple
Criterio 1	Se entrega un proyecto de Terraform.	1	10%	Sí
Criterio 2	<i>Security Groups</i> .	0,5	5%	Sí
Criterio 3	El <i>Provider</i> de Terraform es correcto.	2	20%	Sí
Criterio 4	Existen dos instancias y se demuestra su despliegue en el PDF.	2	20%	Sí
Criterio 5	Las dos instancias están configuradas y se muestra el <i>Connection String</i> a MongoDB.	2	20%	Sí
Criterio 6	Dos instancias con balanceo de tráfico.	1	10%	Sí
Criterio 7	Explicación del vídeo completa.	1,5	15%	Sí
		10	100 %	

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Creación de usuario:

Se crea un usuario IAM, para mayor seguridad y no trabajar con el usuario raíz.

The screenshot shows the AWS IAM console page for a user named 'bricexgeo'. The page has a breadcrumb trail: IAM > Usuarios > bricexgeo. There is an 'Eliminar' button in the top right. The main content area is titled 'bricexgeo Información'. Below this is a 'Resumen' section with three columns of information:

- ARN:** `arn:aws:iam::533267433227:user/bricexgeo`
- Acceso a la consola:** **Habilitado sin MFA** (with a warning icon)
- Clave de acceso 1:** `AKIAXYKJXI4F5U74GIGW` - Active. **Usado hoy, 13 horas antiguo.**
- Creado:** January 25, 2024, 23:45 (UTC-05:00)
- Último inicio de sesión en la consola:** **Nunca** (with a clock icon)
- Clave de acceso 2:** [Crear clave de acceso](#)

Below the summary are tabs for 'Permisos', 'Grupos (1)', 'Etiquetas (1)', 'Credenciales de seguridad', and 'Access Advisor'. The 'Permisos' tab is active, showing 'Políticas de permisos (1)'. It includes a search bar, a 'Filtrar por Tipo' dropdown (set to 'Todos los tipos'), and a table of permissions:

<input type="checkbox"/>	Nombre de la política	Tipo	Adjuntado a través de
<input type="checkbox"/>	AdministratorAccess	Administrada por AWS: función de trabajo	Grupo admin

Creación de grupo de seguridad:

Se añade el usuario creado a un grupo de usuario con los permisos correspondientes.

The screenshot shows the AWS IAM console page for a group named 'admin'. The page has a breadcrumb trail: IAM > Grupos de usuarios > admin. There is an 'Eliminar' button in the top right. The main content area is titled 'admin Información'. Below this is a 'Resumen' section with three columns of information:

- Nombre del grupo de usuarios:** admin
- Hora de creación:** January 25, 2024, 23:44 (UTC-05:00)
- ARN:** `arn:aws:iam::533267433227:group/admin`

Below the summary are tabs for 'Usuarios (1)', 'Permisos', and 'Access Advisor'. The 'Usuarios (1)' tab is active, showing 'Usuarios de este grupo (1)'. It includes a search bar, a 'Filtrar por Tipo' dropdown (set to 'Todos los tipos'), and a table of users:

<input type="checkbox"/>	Nombre de usuario	Grupos	Última acti...	Hora de creación
<input type="checkbox"/>	bricexgeo	1	Ninguno	hace 13 horas

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Creación de par de claves:

Se crea un par de clave, para tener el archivo .pem.

Pares de claves (1) Información					
<input type="text" value="Buscar Key Pair por atributo o etiqueta"/>					
<input type="checkbox"/>	Nombre	Tipo	Se ha creado	Huella digital	ID
<input type="checkbox"/>	passgeobricex	rsa	2024/01/25 23:39 GMT-5	26:38:5b:6e:5b:9d:28:f9:e2:1e:33:b5e...	key-06d56240...

Revisión de la VPC:

Por defecto se debe disponer de una VPC, caso contrario se debe crear una.

Sus VPC (1/1) Información					
<input type="text" value="Search"/>					
<input checked="" type="checkbox"/>	Name	ID de la VPC	Estado	CIDR IPv4	CIDR IPv6
<input checked="" type="checkbox"/>	-	vpc-077486477ed75de11	Available	172.31.0.0/16	-

Detalles	Mapa de recursos	CIDR	Registros de flujo	Etiquetas	Integraciones
----------	------------------	------	--------------------	-----------	---------------

Detalles			
ID de la VPC vpc-077486477ed75de11	Estado Available	Nombres de host de DNS Habilitado	Resolución de DNS Habilitado
Tenencia Default	Conjunto de opciones de DHCP dopt-028e1f46b191491e8	Tabla de enrutamiento principal rtb-0b1f27ea369386eb5	ACL de red principal acl-082c58f9ebcc4b966
VPC predeterminada Sí	CIDR IPv4 172.31.0.0/16	Grupo IPv6 -	CIDR IPv6 (grupo de bordes de red) -
Métricas de uso de direcciones de red Desactivado	Grupos de reglas del firewall de DNS de Route 53 Resolver -	ID de propietario 533267433227	

Creación de grupos de seguridad:

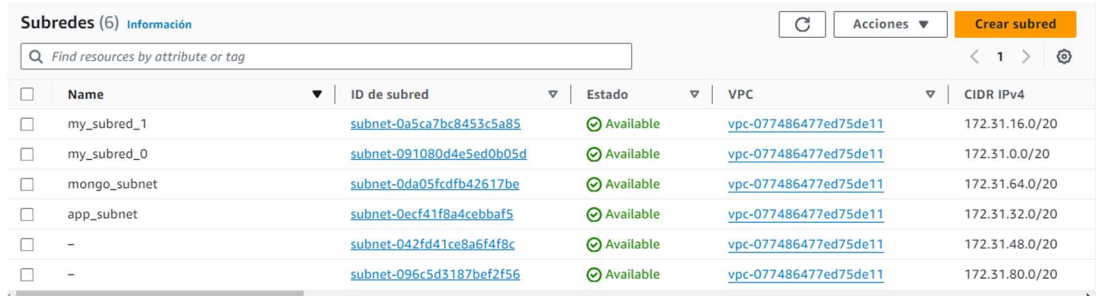
Se debe tener un grupo de seguridad, en este caso se lo crea mediante terraform.

Grupos de seguridad (4) Información				
<input type="text" value="Find resources by attribute or tag"/>				
<input type="checkbox"/>	Name	Security group ID	Nombre del grupo de seguridad	ID de la VPC
<input type="checkbox"/>	my_securitygroup	sg-08efbe1a87b8e473a	my_securitygroup	vpc-077486477ed75de11
<input type="checkbox"/>	-	sg-02a468b15362cf0a9	app_sg	vpc-077486477ed75de11
<input type="checkbox"/>	-	sg-0e69d1d0f7710f	default	vpc-077486477ed75de11
<input type="checkbox"/>	-	sg-0c172c38d0a32f4d9	mongo_sg	vpc-077486477ed75de11

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Creación de subredes:

De igual forma se debe tener las subredes configuradas, en este caso la creación y configuración se realiza con terraform.



The screenshot shows the AWS Subnets console with a table of subnets. The table has columns for Name, ID de subred, Estado, VPC, and CIDR IPv4. There are 6 subnets listed, all with an 'Available' status and associated with the 'vpc-077486477ed75de11' VPC.

Name	ID de subred	Estado	VPC	CIDR IPv4
my_subred_1	subnet-0a5ca7bc8453c5a85	Available	vpc-077486477ed75de11	172.31.16.0/20
my_subred_0	subnet-091080d4e5ed0b05d	Available	vpc-077486477ed75de11	172.31.0.0/20
mongo_subnet	subnet-0da05fcdfb42617be	Available	vpc-077486477ed75de11	172.31.64.0/20
app_subnet	subnet-0ecf41f8a4cebbaf5	Available	vpc-077486477ed75de11	172.31.32.0/20
-	subnet-042fd41ce8a6f4f8c	Available	vpc-077486477ed75de11	172.31.48.0/20
-	subnet-096c5d3187bef2f56	Available	vpc-077486477ed75de11	172.31.80.0/20

Contenido del archivo node:

Es la configuración que tendrá el nginx.

```
server {
    listen 80;

    server_name _;

    location / {
        proxy_pass http://localhost:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Contenido del archivo app_config:

Se realizará las instalaciones y configuraciones necesarias en nuestra instancia donde estará la aplicación.

```
#!/usr/bin/env bash
sleep 30
```

```
# Install node js
curl -fsSL https://deb.nodesource.com/setup_15.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo apt install build-essential -y
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Install pm2

```
sudo npm install -g pm2
```

Install NGINX

```
sudo apt update -y
```

```
sudo apt-get install nginx -y
```

```
systemctl enable nginx
```

```
sudo rm /etc/nginx/sites-enabled/default
```

```
sudo mv /tmp/node /etc/nginx/sites-available/node
```

```
sudo ln -s /etc/nginx/sites-available/node /etc/nginx/sites-enabled/node
```

```
# sudo systemctl restart nginx
```

```
service nginx restart
```

Setup firewall

```
sudo ufw allow ssh
```

```
sudo ufw allow http
```

```
sudo ufw allow https
```

```
ufw enable
```

Configure pm2 to run app on startup

```
mkdir -p ~/code/app-dist
```

```
mv /tmp/hello.js ~/code/app-dist/hello.js
```

```
cd ~/code/app-dist/
```

Instalar dotenv y mongodb npm packages

```
sudo npm init -y # Crea un package.json si no existe
```

```
sudo npm install dotenv mongodb
```

```
sudo apt install mongodb-clients
```

#Iniciar el Node.js

```
sudo pm2 start hello.js
```

```
sudo pm2 startup systemd
```

```
sudo pm2 save
```

```
sudo pm2 list
```

```
sudo shutdown -r now
```

```
exit 0
```

Contenido del archivo bd_config:

Se realizará las instalaciones y configuraciones necesarias en nuestra instancia donde estará la base de datos.

```
#!/usr/bin/env bash
```

```
sleep 30
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```
sudo apt-get update
```

```
# Install mongo
```

```
sudo apt-get install -y mongodb
```

```
mongo --eval 'db.createUser({user: "my_user", pwd: "my_password",
roles: [{role: "readWrite", db: "my_db"}]})'
```

```
mongo my_db --eval 'db.createCollection("my_collection")'
```

```
# Ruta al archivo de configuración de MongoDB
```

```
CONFIG_FILE="/etc/mongodb.conf"
```

```
# Realiza una copia de seguridad del archivo de configuración original
```

```
sudo cp $CONFIG_FILE ${CONFIG_FILE}.bak
```

```
# Cambia '127.0.0.1' por '0.0.0.0' en la configuración de 'bindIp'
```

```
sudo sed -i 's/bindIp: 127.0.0.1/bindIp: 0.0.0.0/' $CONFIG_FILE
```

```
# Reinicia el servicio MongoDB para aplicar los cambios
```

```
sudo systemctl restart mongodb
```

```
echo "Configuración de MongoDB actualizada y servicio reiniciado."
```

```
exit 0
```

Contenido del archivo variables.tf:

En este archivo se almacena todas las variables que se utilizará para la plantilla del main.tf

```
# ----- CREDENCIALES AWZ -----
```

```
# Variables de llave acceso
```

```
variable "aws_access_key" {
  type    = string
  default = ""
}
```

```
# Variables de llave acceso secreta
```

```
variable "aws_secret_key" {
  type    = string
  default = ""
}
```

```
# Variables de llave acceso secreta
```

```
variable "region" {
  type    = string
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```

default = "us-east-1"
}

# Nombre de pares de clave
variable "key_name" {
  type    = string
  default = "passgeobricex"
}

# ID de la VPC
variable "vpc_id" {
  type    = string
  default = "vpc-077486477ed75de11"
}

# Configuración de las subredes
variable "subnets" {
  default = [
    { cidr_block = "172.31.0.0/20", az = "us-east-1a" },
    { cidr_block = "172.31.16.0/20", az = "us-east-1c" },
  ]
}

# ----- FIN CREDENCIALES AWZ -----

# ----- VARIABLES DE MONGODB -----

# IP privada .16
variable "mongo_priv_ip" {
  type    = string
  default = "172.31.0.8"
}

# ----- FIN VARIABLES DE MONGODB -----

# ----- VARIABLES DE APLICACIÓN -----

# IP privada .32
variable "app_priv_ip" {
  type    = string
  default = "172.31.0.24"
}

variable "app_priv_ip_2" {
  type    = string
  default = "172.31.16.48"
}

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

----- FIN VARIABLES DE APLICACIÓN -----

Contenido del archivo main.tf:

En este archivo main.tf es la plantilla principal, el cual se encargará de toda la ejecución correcta, para cumplir con los criterios de la actividad.

Bloque de configuración de Terraform

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws" # Indica que se utilizará el proveedor
      AWS de HashiCorp.
      version = "~> 3.27"      # Especifica que se debe usar una versión
      del proveedor AWS que sea compatible con la versión 3.27, pero no
      una versión mayor que 4.0.
    }
  }
  required_version = ">= 0.14.9" # Especifica que esta configuración de
  Terraform requiere al menos la versión 0.14.9 de Terraform.
}
```

Configuración del proveedor AWS

```
provider "aws" {
  profile = "default"      # Utiliza el perfil "default" definido en las
  credenciales de AWS.
  region = var.region      # Establece la región de AWS a la que se
  conectará Terraform.
  access_key = var.aws_access_key # La clave de acceso de AWS para
  la autenticación.
  secret_key = var.aws_secret_key # La clave secreta de AWS para la
  autenticación.
}
```

Creación de un grupo de seguridad en AWS

```
resource "aws_security_group" "my_securitygroup" {
  description = "Grupo de seguridad" # Descripción del grupo de
  seguridad.
  name = "my_securitygroup" # Nombre asignado al grupo de
  seguridad.
  vpc_id = var.vpc_id      # ID de la VPC donde se creará el grupo
  de seguridad.
}
```

Reglas de entrada

```
ingress {
  from_port = 0      # Desde el puerto (0 indica todos los puertos)
  to_port = 0        # Hasta el puerto (0 indica todos los puertos)
}
```


Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```

protocol = "-1"          # Protocolo (-1 indica todos los protocolos)
cidr_blocks = ["0.0.0.0/0"] # Bloques CIDR desde donde se permite el
tráfico entrante.
}

```

```

# Reglas de salida
egress {
  from_port = 0          # Desde el puerto (0 indica todos los puertos)
  to_port   = 0          # Hasta el puerto (0 indica todos los puertos)
  protocol  = "-1"       # Protocolo (-1 indica todos los protocolos)
  cidr_blocks = ["0.0.0.0/0"] # Bloques CIDR a los que se permite el
tráfico saliente.
}

```

```

tags = {
  Name = "my_securitygroup" # Etiqueta para identificar el grupo de
seguridad.
}
}

```

```

resource "aws_subnet" "my_subred" {
  count          = length(var.subnets)
  vpc_id         = var.vpc_id
  cidr_block     = var.subnets[count.index].cidr_block
  availability_zone = var.subnets[count.index].az

  tags = {
    Name = "my_subred_${count.index}"
  }
}

```

Obtención de una AMI de Ubuntu

```

data "aws_ami" "ubuntu" {
  most_recent = true          # Indica que se debe obtener la AMI más
reciente.
  owners      = ["099720109477"] # ID del propietario de la AMI, en este
caso Canonical.

```

```

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/ubuntu-*20.04-amd64-server-*"] #
Filtrar las AMI para obtener una versión específica de Ubuntu.
  }
}

```

Configuración de una instancia EC2 para MongoDB

```

resource "aws_instance" "mongodb" {

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```

ami = data.aws_ami.ubuntu.id # ID de la AMI a
usar para la instancia.
instance_type = "t2.micro" # Tipo de
instancia EC2.
key_name = var.key_name # Nombre del
key pair para acceso SSH.
subnet_id = aws_subnet.my_subnet[0].id # Subred
donde se lanzará la instancia de MongoDB (subred privada).
vpc_security_group_ids =
[aws_security_group.my_securitygroup.id] # ID del grupo de seguridad
asociado.
private_ip = var.mongo_priv_ip # Dirección IP
privada asignada a la instancia.
associate_public_ip_address = true

tags = {
  Name = "Servidor_MongoDB" # Etiqueta para identificar la instancia
de MongoDB.
}

# Provisioner para copiar un archivo de configuración a la instancia
provisioner "file" {
  source = "app/bd_config.sh" # Ruta local del archivo de
configuración
  destination = "/tmp/bd_config.sh" # Ruta de destino en la instancia

  # Configuración de la conexión SSH para transferir el archivo
  connection {
    type = "ssh"
    user = "ubuntu"
    private_key = file("passgeobricex.pem")
    host = self.public_ip
  }
}

provisioner "remote-exec" {
  inline = [
    "ls -l /tmp/bd_config.sh", # Lista el archivo bd_config.sh para
verificar su existencia y permisos
    "chmod +x /tmp/bd_config.sh", # Cambia los permisos del archivo
para hacerlo ejecutable
    "/tmp/bd_config.sh" # Ejecuta el script bd_config.sh
  ]

  connection {
    type = "ssh"
    user = "ubuntu"
  }
}

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```

    private_key = file("passgeobricex.pem")
    host       = self.public_ip # Dirección IP pública
  }
}
}

resource "aws_instance" "app_server" {
  count          = 2 # Crea dos instancias
  ami            = data.aws_ami.ubuntu.id
  instance_type  = "t2.micro"
  key_name       = var.key_name
  subnet_id      = aws_subnet.my_subred[count.index %
length(var.subnets)].id
  vpc_security_group_ids =
[aws_security_group.my_securitygroup.id]
  private_ip     = count.index == 0 ? var.app_priv_ip :
var.app_priv_ip_2
  associate_public_ip_address = true
  tags = {
    Name = "Servidor_Web-${count.index + 1}"
  }
}

# Provisioner para crear un archivo hello.js en la instancia
provisioner "file" {
  content = <<-EOT
    const http = require('http');
    const MongoClient = require('mongodb').MongoClient;

    const hostname = 'localhost';
    const port = 8080;

    // Asegúrate de que la dirección IP sea la correcta
    const url =
`mongodb://${aws_instance.mongodb.private_ip}:27017/my_bd`;

    let dbConnection;

    MongoClient.connect(url, { useNewUrlParser: true,
useUnifiedTopology: true }, (err, client) => {
      if (err) {
        console.error('Error al conectar a MongoDB:', err);
      } else {
        console.log('Conectado exitosamente a MongoDB');
        dbConnection = client.db('my_bd');
        // Aquí puedes hacer algo con la conexión a MongoDB
      }
    });
  >>>EOT
}

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain; charset=utf-8');

  let message = "¡HOLA MUNDO! ¡Soy Brito Casanova
Geovanny!\n";
  if (dbConnection) {
    message += "Conectado exitosamente a MongoDB en " + url;
  } else {
    message += "No se pudo conectar a MongoDB.";
  }
  res.end(message);
});

server.listen(port, hostname, () => {
  console.log("Servidor corriendo en http://" + hostname + ":" +
port + "/");
});

```

EOT

#Destino del fichero

destination = "/tmp/hello.js"

```

connection {
  type      = "ssh"
  user      = "ubuntu"
  private_key = file("passgeobricex.pem")
  host      = self.public_ip
}
}

```

Provisioner para copiar un archivo de configuración a la instancia

```

provisioner "file" {
  source      = "app/app_config.sh" # Ruta local del archivo de
configuración
  destination = "/tmp/app_config.sh" # Ruta de destino en la instancia

```

Configuración de la conexión SSH para transferir el archivo

```

connection {
  type      = "ssh"
  user      = "ubuntu"
  private_key = file("passgeobricex.pem")
  host      = self.public_ip
}
}

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Un provisioner 'file' que copia un archivo local a una ubicación remota en la instancia

```
provisioner "file" {
  source      = "app/node" # Ruta del archivo local a copiar
  destination = "/tmp/node" # Ruta de destino en la instancia remota
  # Configuración de conexión SSH para la instancia remota
  connection {
    type      = "ssh"          # Tipo de conexión, en este caso SSH
    user      = "ubuntu"       # Nombre de usuario para la
conexión SSH
    private_key = file("passgeobricex.pem") # Llave privada para la
conexión SSH
    host        = self.public_ip # Dirección IP pública de la instancia
(obtenida dinámicamente)
  }
}
```

Un provisioner 'remote-exec' que ejecuta comandos en la instancia remota

```
provisioner "remote-exec" {
  # Lista de comandos que se ejecutarán en la instancia remota
  inline = [
    "ls -l /tmp/app_config.sh", # Lista el archivo app_config.sh para
verificar su existencia y permisos
    "chmod +x /tmp/app_config.sh", # Cambia los permisos del archivo
para hacerlo ejecutable
    "/tmp/app_config.sh"         # Ejecuta el script app_config.sh
  ]
  # Configuración de conexión SSH, similar al provisioner 'file'
  connection {
    type      = "ssh"
    user      = "ubuntu"
    private_key = file("passgeobricex.pem")
    host        = self.public_ip
  }
}
```

Creación del Application Load Balancer (ALB)

```
resource "aws_lb" "my_alb" {
  name      = "hello-mean" # Nombre del
ALB
  internal  = false        # Especifica que el
ALB es accesible desde Internet
  load_balancer_type = "application" # Tipo de
ALB, en este caso, un Application Load Balancer
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```

security_groups =
[aws_security_group.my_securitygroup.id]          # Grupos de
seguridad asociados al ALB
subnets        = [aws_subnet.my_subred[0].id,
aws_subnet.my_subred[1].id] # Subredes donde se desplegará el ALB

enable_deletion_protection = false # Desactiva la protección contra
borrado para permitir eliminar el ALB

```

```

tags = {
  Name = "my-alb" # Etiqueta para identificar el ALB
}

```

Creación del Target Group para el ALB

```

resource "aws_lb_target_group" "my_target_group" {
  name     = "my-target-group" # Nombre del Target Group
  port     = 80                # Puerto en el que el Target Group recibirá tráfico
  protocol = "HTTP"            # Protocolo utilizado por el Target Group
  vpc_id   = var.vpc_id        # ID de la VPC en la que se crea el Target
Group

```

Configuración del chequeo de salud

```

health_check {
  healthy_threshold = 2    # Número de chequeos exitosos para
considerar una instancia sana
  unhealthy_threshold = 2  # Número de chequeos fallidos para
considerar una instancia insana
  timeout           = 3    # Tiempo de espera para cada chequeo de
salud
  path              = "/"  # Ruta utilizada para el chequeo de salud
  protocol          = "HTTP" # Protocolo utilizado para el chequeo de
salud
  matcher           = "200" # Código de respuesta esperado para
considerar sana una instancia
  interval          = 30   # Intervalo entre chequeos de salud
}

```

```

resource "aws_lb_target_group_attachment" "attach_instance" {
  count          = 2
  target_group_arn = aws_lb_target_group.my_target_group.arn
  target_id      = aws_instance.app_server.*.id[count.index]
  port           = 80
}

```

Creación del Listener para el ALB

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

```
resource "aws_lb_listener" "my_listener" {
  load_balancer_arn = aws_lb.my_alb.arn # ARN del ALB
  port              = "80"              # Puerto en el que el Listener escuchará
  protocol          = "HTTP"           # Protocolo utilizado por el Listener

  # Acción por defecto del Listener
  default_action {
    type = "forward" # Tipo de acción, en este
caso, reenviar el tráfico
    target_group_arn = aws_lb_target_group.my_target_group.arn # ARN
del Target Group al que se reenvía el tráfico
  }
}
```

Logs:

terraform init:

Inicializa un directorio de trabajo de Terraform, descargando e instalando módulos y proveedores necesarios para el proyecto.

```
PS D:\PROGRAMACION\Vagrant\terraform-mean-lasted> terraform init
●
Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.75.2

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

terraform fmt:

Revisa y reescribe archivos de configuración de Terraform para ajustarse al formato estándar y a las convenciones de estilo.

```
PS D:\PROGRAMACION\Vagrant\terraform-mean-lasted> terraform fmt
● main.tf
```

terraform validate:

Valida la configuración de Terraform para comprobar errores de sintaxis y que todos los atributos y referencias sean correctos.

```
● PS D:\PROGRAMACION\Vagrant\terraform-mean-lasted> terraform validate
● Success! The configuration is valid.
```


Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

terraform plan:

Crea un plan de ejecución, mostrando qué acciones se realizarán en la infraestructura al aplicar la configuración de Terraform.

```
PS D:\PROGRAMACION\Vagrant\terraform-mean-lasted> terraform plan
data.aws_ami.ubuntu: Reading...
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-01206169971a6406e]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-055c25ef7e639154a]
aws_lb_target_group.my_target_group: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:533267433227:targetgroup/my-target-group/aa8adcb741427914]
aws_security_group.my_securitygroup: Refreshing state... [id=sg-04b7b344da137ab8b]
aws_lb.my_alb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:533267433227:loadbalancer/app/hello-mean/f8f609b0ad21b204]
data.aws_ami.ubuntu: Read complete after 1s [id=ami-0d617a6646547e912]
aws_instance.mongodb: Refreshing state... [id=i-0819df929f0eb7e38]
aws_instance.app_server[0]: Refreshing state... [id=i-0a187962d98a4bd0c]
aws_instance.app_server[1]: Refreshing state... [id=i-0f3097f07a7405250]
aws_lb_target_group_attachment.attach_instance[0]: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:533267433227:targetgroup/my-target-group/aa8adcb741427914-20240126193608899000000002]
aws_lb_target_group_attachment.attach_instance[1]: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:533267433227:targetgroup/my-target-group/aa8adcb741427914-20240126193608899000000001]
aws_lb_listener.my_listener: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:533267433227:listener/app/hello-mean/f8f609b0ad21b204/374e9f06926b9718]
```

```
# aws_instance.app_server[0] will be created
+ resource "aws_instance" "app_server" {
+   ami               = "ami-0d617a6646547e912"
+   arn               = (known after apply)
+   associate_public_ip_address = true
+   availability_zone = (known after apply)
+   cpu_core_count    = (known after apply)
+   cpu_threads_per_core = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized      = (known after apply)
+   get_password_data   = false
+   host_id            = (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           = "passgeobricex"
+   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)
+   ami               = "ami-0d617a6646547e912"
+   arn               = (known after apply)
+   associate_public_ip_address = true
+   availability_zone = (known after apply)
+   cpu_core_count    = (known after apply)
+   cpu_threads_per_core = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized      = (known after apply)
+   get_password_data   = false
+   host_id            = (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           = "passgeobricex"
+   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)
```

terraform apply:

Aplica los cambios descritos por terraform plan para alcanzar el estado deseado de la infraestructura definida en la configuración.

```
+ resource "aws_subnet" "my_subnet" {
+   arn               = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone = "us-east-1c"
+   availability_zone_id = (known after apply)
+   cidr_block         = "172.31.16.0/20"
+   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 = false
+   owner_id           = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   tags               = {
+     "Name" = "my_subnet_1"
+   }
+   tags_all          = {
+     "Name" = "my_subnet_1"
+   }
+   vpc_id            = "vpc-077486477ed75de11"
+ }

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
```


Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

[illegible]

Crea automáticamente el grupo de seguridad a usar.

Nombre del grupo de seguridad my_securitygroup	ID del grupo de seguridad sg-0a5727e6bb0d2b871	Descripción Grupo de seguridad	ID de la VPC vpc-077486477ed75de11 🔗
Propietario 533267433227	Número de reglas de entrada 1 Entrada de permiso	Número de reglas de salida 1 Entrada de permiso	

Crea automáticamente las subredes necesarias a usar.

<p>ID de subred</p> <p> subnet-0e70aca93e57f4511</p> <p>Direcciones IPv4 disponibles</p> <p> 4091</p> <p>Grupo de borde de red</p> <p>-</p> <p>Subred predeterminada</p> <p>No</p> <p>Grupo IPv4 propiedad del cliente</p> <p>-</p>	<p>ARN de subred</p> <p> arn:aws:ec2:us-east-1:533267433227:subnet/subnet-0e70aca93e57f4511</p> <p>CIDR IPv6</p> <p>-</p> <p>VPC</p> <p>vpc-077486477ed75de11</p> <p>Asignación automática de la dirección IPv4 pública</p> <p>No</p>	<p>Estado</p> <p> Available</p> <p>Zona de disponibilidad</p> <p> us-east-1c</p> <p>Tabla de enrutamiento</p> <p>rtb-0b1f27ea369386eb5</p> <p>Asignar dirección IPv6 automáticamente</p> <p>No</p> <p>Reservas de CIDR 4 IPv</p>	<p>CIDR IPv4</p> <p> 172.31.16.0/20</p> <p>ID de zona de disponibilidad</p> <p> use1-az4</p> <p>ACL de red</p> <p>acl-082c58f9ebcc4b966</p> <p>Asignación automática de direcciones IPv4 propiedad del cliente</p> <p>No</p> <p>Reservas de CIDR 6 IPv</p>
---	--	--	--

Crea las instancias necesarias según la configuración.

Instancias (1/3) Información

Any state

Estado de la instancia: running

Quitar los filtros

	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de	Estado de la al...	Zona d
<input type="checkbox"/>	Servidor_Web-2	i-04cfc509cf80c21c	En ejecución	t2.micro	2 Inicializando	View alarms	us-east
<input type="checkbox"/>	Servidor_MongoDB	i-Deefdaed44b32455c	En ejecución	t2.micro	2/2 comprobador	View alarms	us-east
<input checked="" type="checkbox"/>	Servidor_Web-1	i-0a49fa4fbd878e1f3	En ejecución	t2.micro	2 Inicializando	View alarms	us-east

Instancia: i-0a49fa4fbd878e1f3 (Servidor_Web-1)

Detalles

Status and alarms New

Monitoreo

Seguridad

Redes

Almacenamiento

Etiquetas

▼ Resumen de instancia Información

ID de la instancia

i-0a49fa4fbd878e1f3 (Servidor_Web-1)

Dirección IPv4 pública

3.235.153.67 [dirección abierta](#)

Estado de la instancia

En ejecución

Direcciones IPv4 privadas

172.31.0.24

DNS de IPv4 pública

ec2-3-235-153-67.compute-1.amazonaws.com

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Crea el balanceador de carga.

Balanceadores de carga (1/1)

Elastic Load Balancing escala automáticamente la capacidad del equilibrador de carga en respuesta a los cambios en el tráfico entrante.

Buscar:

Nombre	Nombre de DNS	Estado	ID de VPC	Zonas de disponibil...	Tipo
hello-mean	hello-mean-795884884.us...	Activo	vpc-077486477ed75d...	2 Zonas de disponibilidad	appli

Equilibrador de carga: hello-mean

Tipo de equilibrador de carga	Estado	VPC	Tipo de dirección IP
Aplicación	Activo	vpc-077486477ed75de11	IPv4
Esquema	Zona hospedada	Zonas de disponibilidad	Fecha creada
Internet-facing	Z35SXDOTRQ7X7K		26 de enero de 2024, 15:01 (UTC-05:00)

Resultados:

Comprobación de conexión manual desde la instancia Servidor_Web-1 a Servidor_MongoDB

```
ubuntu@ip-172-31-0-24:~$ nc -vz 172.31.0.8 27017
Connection to 172.31.0.8 27017 port [tcp/*] succeeded!

ubuntu@ip-172-31-0-24:~$ mongo --host 172.31.0.8 --port 27017
MongoDB shell version v3.6.8
connecting to: mongodb://172.31.0.8:27017/
Implicit session: session { "id" : UUID("23e1d18d-88af-47b7-b6e2-508528fc723f") }
MongoDB server version: 3.6.8
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
http://docs.mongodb.org/
Questions? Try the support group
http://groups.google.com/group/mongodb-user
Server has startup warnings:
2024-01-26T20:03:09.474+0000 I STORAGE [initandlisten]
2024-01-26T20:03:09.474+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2024-01-26T20:03:09.474+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-01-26T20:03:10.761+0000 I CONTROL [initandlisten]
2024-01-26T20:03:10.761+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2024-01-26T20:03:10.761+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2024-01-26T20:03:10.761+0000 I CONTROL [initandlisten]
```

Instancia 1:

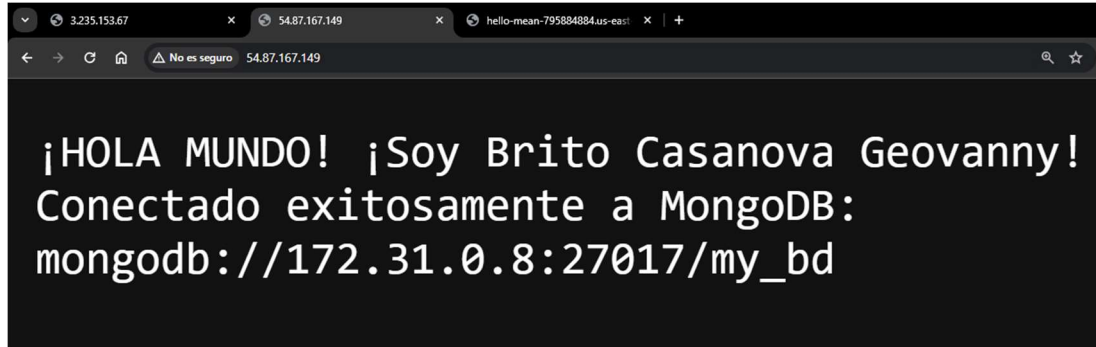
Se ha levantado la primera instancia en la subred 172.31.0.24, en us-east-1a la cual fue creada automáticamente desde terraform.

¡HOLA MUNDO! ¡Soy Brito Casanova Geovanny!
Conectado exitosamente a MongoDB:
mongodb://172.31.0.8:27017/my_bd

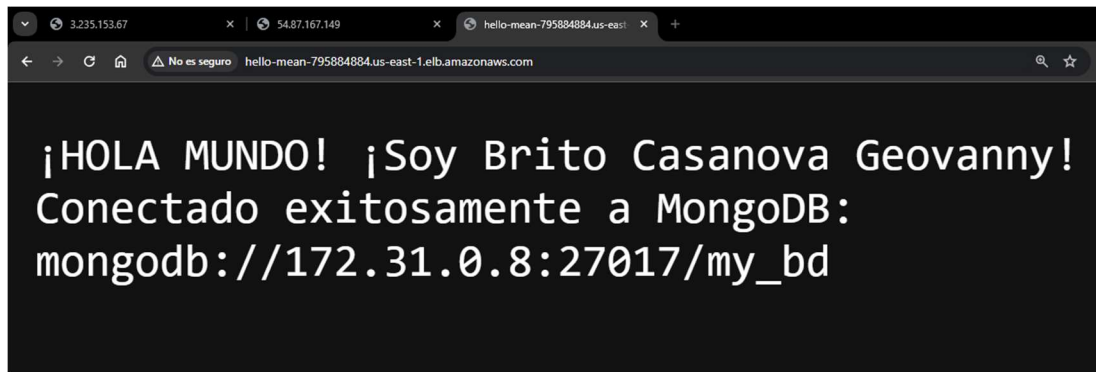
Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Instancia 2:

Se ha levantado la primera instancia en la subred 172.31.16.24, en us-east-1c, la cual fue creada automáticamente desde terraform.



Balanceador de carga:



Video:

<https://youtu.be/Nz1g2c4wj6I>

Resumen:

En este proyecto, se utilizó Terraform para automatizar el despliegue de una stack MEAN en AWS, mostrando el poder y la flexibilidad de la infraestructura como código. Se crearon tres instancias EC2 en una VPC: dos para alojar un servidor Node.js con Nginx y otra para la base de datos MongoDB.

El servidor de aplicaciones Node.js fue configurado para servir una aplicación que devuelve un mensaje junto con la cadena de conexión a la base de datos MongoDB. La instancia de MongoDB fue asegurada con un grupo de seguridad que permite el tráfico entrante en el puerto 27017 solo desde la instancia de la aplicación, garantizando así una comunicación segura entre los servicios.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Brito Casanova.	24/01/2024
	Nombre: Geovanny José.	

Conclusiones:

A lo largo de este proyecto, se ha demostrado que Terraform es una herramienta indispensable para la implementación y gestión de infraestructura como código. Con Terraform, se logró desplegar una stack MEAN en AWS de manera eficiente, lo que subraya la capacidad de Terraform para crear infraestructuras complejas y altamente disponibles. La automatización de este proceso no solo ahorra tiempo, sino que también minimiza los errores humanos, proporcionando una base sólida y confiable para el despliegue de aplicaciones.

El uso de Terraform para definir la infraestructura requerida por la stack MEAN permitió establecer una práctica de integración continua para la infraestructura, similar a lo que se hace con el código de la aplicación. Cada componente, desde la VPC y las instancias EC2 hasta los grupos de seguridad, fue configurado con precisión, demostrando el control detallado que Terraform otorga sobre los recursos en la nube. La consistencia obtenida a través de este enfoque garantiza que la infraestructura pueda ser replicada en diferentes ambientes, facilitando las pruebas y la escalabilidad.

Finalmente, este proyecto sirvió como una plataforma de aprendizaje significativa sobre la implementación de prácticas de DevOps en la nube. Se reforzó el concepto de que una infraestructura bien planificada y ejecutada es tan crítica como el código que corre sobre ella. Al concluir este proyecto, queda claro que la habilidad para orquestar y manejar la infraestructura de la nube de forma programática no es solo una habilidad deseable, sino una necesidad en el mundo moderno del desarrollo y operaciones de sistemas.