

Implementación de Servicios Básicos en AWS con Terraform

[Configuración y Código Terraform](#)

- [1. Archivo main.tf](#)
- [2. Archivo variables.tf](#)
- [3. Archivo outputs.tf](#)
- [4. Capturas de correcto funcionamiento](#)

Configuración y Código Terraform

1. Archivo main.tf

Se utilizó el siguiente archivo de Terraform para definir los recursos:

```
# Configuración del proveedor de AWS
provider "aws" {
  region = var.aws_region # Define la región de AWS que se utilizará
}

# Obtener la identidad de la cuenta actual en AWS
data "aws_caller_identity" "current" {}

# Crear un grupo de seguridad para la instancia EC2
resource "aws_security_group" "web_sg" {
  name      = "web_sg" # Nombre del grupo de seguridad
  description = "Permitir tráfico HTTP SSH" # Descripción del propósito del grupo

  # Permitir tráfico SSH (puerto 22) desde cualquier IP
  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Permitir tráfico HTTP (puerto 80) desde cualquier IP
  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Permitir todo el tráfico saliente (egress)
  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Etiquetas para identificar el grupo de seguridad
}
```

```

tags = {
  Name = "web_sg"
  Owner = "Alexandro Surraco"
}
}

# Crear un rol de IAM para que la instancia EC2 acceda a S3
resource "aws_iam_role" "ec2_role" {
  name = "ec2_s3_access_role" # Nombre del rol IAM

  # Política que permite a EC2 asumir este rol
  assume_role_policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Effect = "Allow",
        Principal = {
          Service = "ec2.amazonaws.com" # Define que EC2 puede asumir este rol
        },
        Action = "sts:AssumeRole"
      }
    ]
  })
}

# Crear una política de IAM para permitir acceso al bucket S3
resource "aws_iam_policy" "s3_access_policy" {
  name = "s3_access_policy" # Nombre de la política IAM

  # Define los permisos específicos para S3
  policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Effect = "Allow", # Permitir las acciones
        Action = [
          "s3:ListBucket" # Permiso para listar objetos en el bucket
        ],
        Resource = [
          "arn:aws:s3:::desafio12-bucket-asurraco" # Aplica a este bucket
        ]
      },
      {
        Effect = "Allow", # Permitir la descarga de objetos
        Action = [
          "s3:GetObject" # Permiso para obtener objetos del bucket
        ],

```

```

    Resource = [
        "arn:aws:s3:::desafio12-bucket-asurraco/*" # Aplica a los objetos dentro del bucket
    ]
}
]
}))
}

```

Asociar la política de S3 al rol de IAM

```

resource "aws_iam_role_policy_attachment" "ec2_role_s3_policy" {
    role      = aws_iam_role.ec2_role.name # Rol al que se asocia la política
    policy_arn = aws_iam_policy.s3_access_policy.arn # ARN de la política IAM
}

```

Crear un perfil de instancia para vincular el rol a EC2

```

resource "aws_iam_instance_profile" "ec2_instance_profile" {
    name = "ec2_instance_profile" # Nombre del perfil de instancia
    role = aws_iam_role.ec2_role.name # Asociar el rol IAM
}

```

Crear una instancia EC2

```

resource "aws_instance" "web" {
    ami          = "ami-0c02fb55956c7d316" # Imagen AMI de Amazon Linux 2
    instance_type = var.instance_type # Tipo de instancia (ej. t2.micro)
    iam_instance_profile = aws_iam_instance_profile.ec2_instance_profile.name # Perfil de
    instancia para acceso IAM
}

```

```

    vpc_security_group_ids = [aws_security_group.web_sg.id] # Grupo de seguridad
    asociado
}

```

Etiquetas para identificar la instancia

```

tags = {
    Name = "Desafio12-EC2"
    Owner = "Alexandro Surraco"
}

```

Script para configurar la instancia al iniciarse

```

user_data = <<-EOT
#!/bin/bash
yum update -y # Actualizar paquetes
yum install -y httpd aws-cli # Instalar Apache y AWS CLI
systemctl start httpd # Iniciar el servicio Apache
systemctl enable httpd # Configurar Apache para iniciar automáticamente
echo "<h1>Apache esta levantado!</h1>" > /var/www/html/index.html # Crear página
de prueba

```

Configurar y montar el volumen EBS

```

mkfs.ext4 /dev/xvdf # Formatear el volumen como ext4
mkdir -p /desafios # Crear el directorio de montaje
mount /dev/xvdf /desafios # Montar el volumen en el directorio
echo "/dev/xvdf /desafios ext4 defaults 0 0" >> /etc/fstab # Registrar el montaje en fstab

# Asignar permisos adecuados al directorio /desafios
chown ec2-user:ec2-user /desafios
chmod 755 /desafios

# Descargar el archivo desde S3
aws s3 cp s3://desafio12-bucket-asurraco/desafio12.pdf /desafios/ >>
/var/log/user-data.log 2>&1
if [ $? -eq 0 ]; then
    echo "Archivo descargado exitosamente" >> /var/log/user-data.log
else
    echo "Error al descargar el archivo" >> /var/log/user-data.log
fi
EOT
}

# Crear un bucket S3
resource "aws_s3_bucket" "bucket" {
    bucket = "desafio12-bucket-asurraco" # Nombre único del bucket

    # Etiquetas para identificar el bucket
    tags = {
        Name = "Desafio12-S3"
        Owner = "Alexandro Surraco"
    }
}

# Subir un archivo PDF al bucket S3
resource "aws_s3_object" "pdf_upload" {
    bucket = aws_s3_bucket.bucket.id # Bucket donde se subirá el archivo
    key = "desafio12.pdf" # Nombre del archivo en S3
    source = "./Desafio12.pdf" # Ruta del archivo local
}

# Configurar la política del bucket S3
resource "aws_s3_bucket_policy" "bucket_policy" {
    bucket = aws_s3_bucket.bucket.id # Bucket al que se aplica la política

    # Define los permisos de acceso al bucket
    policy = <<POLICY
    {
        "Version": "2012-10-17",
        "Statement": [

```

```

{
  "Sid": "AllowEC2RoleAccess", # Identificador de la política
  "Effect": "Allow",
  "Principal": {
    "AWS":
"arn:aws:iam::${data.aws_caller_identity.current.account_id}:role/${aws_iam_role.ec2_role
.name}" # Rol permitido
  },
  "Action": [
    "s3:ListBucket", # Permitir listar objetos
    "s3:GetObject" # Permitir obtener objetos
  ],
  "Resource": [
    "arn:aws:s3:::${aws_s3_bucket.bucket.id}", # Acceso al bucket
    "arn:aws:s3:::${aws_s3_bucket.bucket.id}/*" # Acceso a los objetos
  ]
}
]
}
POLICY
}

```

Crear un volumen EBS

```

resource "aws_ebs_volume" "volume" {
  availability_zone = "us-east-1a" # Zona de disponibilidad para el volumen
  size              = var.volume_size # Tamaño del volumen (en GB)

```

Etiquetas para identificar el volumen

```

tags = {
  Name   = "Desafio12-EBS"
  Owner  = "Alexandro Surraco"
}
}

```

Adjuntar el volumen EBS a la instancia EC2

```

resource "aws_volume_attachment" "attach" {
  device_name = "/dev/xvdf" # Punto de conexión del volumen
  volume_id   = aws_ebs_volume.volume.id # ID del volumen
  instance_id = aws_instance.web.id # ID de la instancia EC2
}

```

2. Archivo variables.tf

```
variable "aws_region" {  
  description = "Región de AWS" # Define la región donde se desplegarán los recursos  
  default     = "us-east-1" # Valor predeterminado, puede modificarse según necesidad  
}  
  
variable "instance_type" {  
  description = "Tipo de instancia EC2" # Especifica el tipo de instancia EC2  
  default     = "t2.micro" # Valor predeterminado compatible con el Free Tier de AWS  
}  
  
variable "volume_size" {  
  description = "Tamaño del volumen EBS" # Define el tamaño del volumen EBS en GB  
  default     = 2 # Valor predeterminado de 2 GB  
}
```

3. Archivo outputs.tf

```
output "instance_public_ip" {  
  description = "La dirección IP pública de la instancia EC2" # Describe la salida  
  value       = aws_instance.web.public_ip # Devuelve la IP pública de la instancia  
}  
  
output "bucket_name" {  
  description = "El nombre del bucket S3" # Describe la salida  
  value       = aws_s3_bucket.bucket.bucket # Devuelve el nombre del bucket  
}  
  
output "ebs_volume_id" {  
  description = "ID del volumen EBS" # Describe la salida  
  value       = aws_ebs_volume.volume.id # Devuelve el ID del volumen EBS  
}
```

4. Capturas de correcto funcionamiento

```
[ec2-user@ip-172-31-22-251 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M   0    474M   0% /dev
tmpfs           483M   0    483M   0% /dev/shm
tmpfs           483M 440K   483M   1% /run
tmpfs           483M   0    483M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  2.2G   5.9G  27% /
/dev/xvdf       2.0G  6.1M   1.8G   1% /desafios
tmpfs           97M   0     97M   0% /run/user/1000
```

```
[ec2-user@ip-172-31-22-251 ~]$ cd /desafios/
[ec2-user@ip-172-31-22-251 desafios]$ ll
total 88
-rw-r--r-- 1 root root 70014 Dec  3 00:33 desafio12.pdf
```



No es seguro 54.242.31.27

Apache esta levantado!

```
[ec2-user@ip-172-31-22-251 desafios]$ cat /var/log/user-data.log
download: s3://desafio12-bucket-asurraco/desafio12.pdf to desafios/desafio12.pdf
Archivo descargado exitosamente
[ec2-user@ip-172-31-22-251 desafios]$
```

```
[ec2-user@ip-172-31-22-251 desafios]$ aws s3 ls s3://desafio12-bucket-asurraco/
2024-12-03 00:33:58          70014 desafio12.pdf
```