

ANEXO 03 - Infraestructura como código – template VPC_ELB_EC2_RDS

Daniel Romero Sánchez

Grado de ingeniería informática

Administración de redes y sistemas operativos

Jordi Serra Ruiz

J. Ramon Esteban Grifoll

6 de junio del 2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Template utilizado en el TFG nombrado como VPC_ELB_EC2_RDS para desplegar la infraestructura modelo avanzada de forma automatizada.

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  License: 3.0 España de Creative Commons.
  Owner: Daniel Romero Sánchez
  Project: TFG - Infraestructura como código.
  Subject: Administración de redes y sistemas operativos.
  Name: VPC_ELB_EC2_RDS
  Description: 'Despliegue de Infraestructura basica: VPC, ELB, EC2, RDS'

Mappings:
  Variables:
    ProjectName:
      Value: TFG
    Owner:
      Value: 'Daniel Romero Sanchez'

  VpcRange:
    Description: Rango de IP para la VPC
    Value: 192.168.0.0/16

  PublicSubnetRangeA:
    Description: Rango de IP para la subnet pública A
    Value: 192.168.1.0/24

  PublicSubnetRangeB:
    Description: Rango de IP para la subnet pública B
    Value: 192.168.10.0/24

  PrivateSubnetRangeA:
    Description: Rango de IP para la subnet privada A
    Value: 192.168.20.0/24

  PrivateSubnetRangeB:
    Description: Rango de IP para la subnet privada B
    Value: 192.168.30.0/24

#####
##### VPC
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !FindInMap [ Variables, VpcRange, Value ]
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !FindInMap [ Variables, ProjectName, Value ]
        - Key: Owner
          Value: !FindInMap [ Variables, Owner, Value ]

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !FindInMap [ Variables, ProjectName, Value ]
        - Key: Owner
          Value: !FindInMap [ Variables, Owner, Value ]

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC
```

```

PublicSubnetA:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !FindInMap [ Variables, PublicSubnetRangeA, Value ]
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: Public Subnet A

PublicSubnetB:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    CidrBlock: !FindInMap [ Variables, PublicSubnetRangeB, Value ]
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: Public Subnet B

PrivateSubnetA:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !FindInMap [ Variables, PrivateSubnetRangeA, Value ]
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: Private Subnet A

PrivateSubnetB:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    CidrBlock: !FindInMap [ Variables, PrivateSubnetRangeB, Value ]
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: Private Subnet B

NatGatewayEIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc

NatGateway:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGatewayEIP.AllocationId
    SubnetId: !Ref PublicSubnetA

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: Public Routes

DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable

```

```

    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnetARouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnetA

PublicSubnetBRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnetB

PrivateRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: Private Routes

DefaultPrivateRoute:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway

PrivateSubnetARouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable
    SubnetId: !Ref PrivateSubnetA

PrivateSubnetBRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable
    SubnetId: !Ref PrivateSubnetB

NoIngressSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "SG_no-ingress"
    GroupDescription: "Grupo de seguridad Security para la VPC"
    VpcId: !Ref VPC
#####
##### EC2
EC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "SG_EC2"
    GroupDescription: "Grupo de seguridad Security para las EC2"
    VpcId: !Ref VPC
    SecurityGroupIngress:
      # Permitir HTTP
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp: 0.0.0.0/0
      # Permitir SSH
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        CidrIp: 0.0.0.0/0

EC2InstanceA:
  Type: AWS::EC2::Instance

```

```

Properties:
  InstanceType: t3.small
  SecurityGroupIds: [!GetAtt EC2SecurityGroup.GroupId]
  SubnetId: !Ref PublicSubnetA
  KeyName: TFG
  ImageId: ami-043097594a7df80ec
  BlockDeviceMappings:
    -
      DeviceName: /dev/sda1
      Ebs:
        VolumeSize: 8
  UserData:
    Fn::Base64: |
      #!/bin/bash -xe
      yum update -y
      yum install -y httpd
      systemctl start httpd
      systemctl enable httpd
      echo "TFG - CloudFormation - Daniel Romero Sanchez - PETICION A" >
/var/www/html/index.html

EC2InstanceB:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: t3.small
    SecurityGroupIds: [!GetAtt EC2SecurityGroup.GroupId]
    SubnetId: !Ref PublicSubnetB
    KeyName: TFG
    ImageId: ami-043097594a7df80ec
    BlockDeviceMappings:
      -
        DeviceName: /dev/sda1
        Ebs:
          VolumeSize: 8
    UserData:
      Fn::Base64: |
        #!/bin/bash -xe
        yum update -y
        yum install -y httpd
        systemctl start httpd
        systemctl enable httpd
        echo "TFG - CloudFormation - Daniel Romero Sanchez - PETICION B" >
/var/www/html/index.html
#####
##### Loas Balancer
ELBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "SG_ELB"
    GroupDescription: "Grupo de seguridad Security para el ELB"
    VpcId: !Ref VPC
    SecurityGroupIngress:
      # Permitir HTTP
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp: 0.0.0.0/0

GrupoDestinoELB:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: !FindInMap [ Variables, ProjectName, Value ]
    Port: 80
    Protocol: HTTP
    HealthCheckPath: /
    TargetType: instance
    VpcId: !Ref VPC
    HealthCheckIntervalSeconds: 30
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 5

```

```

    UnhealthyThresholdCount: 2
    Matcher:
      HttpStatusCode: 200
    Targets:
      -
        Id: !Ref EC2InstanceA
      -
        Id: !Ref EC2InstanceB

LoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    SecurityGroups: [!GetAtt ELBSecurityGroup.GroupId]
    Subnets: [!Ref PublicSubnetA, !Ref PublicSubnetB]
    Tags:
      - Key: Name
        Value: Load Balancer

ListenerELB:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    LoadBalancerArn: !Ref LoadBalancer
    Port: 80
    Protocol: HTTP
    DefaultActions:
      -
        Type: forward
        TargetGroupArn: !Ref GrupoDestinoELB
#####
##### BBDD
BBDDSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "SG_RDS"
    GroupDescription: "Grupo de seguridad Security para la RDS"
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '3306'
        ToPort: '3306'
        SourceSecurityGroupId: !GetAtt [EC2SecurityGroup, GroupId]

MyDBSubnetGroup:
  Type: AWS::RDS::DBSubnetGroup
  Properties:
    DBSubnetGroupDescription: Grupo de Subnets para RDS
    DBSubnetGroupName: !FindInMap [ Variables, ProjectName, Value ]
    SubnetIds:
      - !Ref PrivateSubnetA
      - !Ref PrivateSubnetB

MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName: !FindInMap [ Variables, ProjectName, Value ]
    DBInstanceIdentifier: !FindInMap [ Variables, ProjectName, Value ]
    VPCSecurityGroups: [!GetAtt [BBDDSecurityGroup, GroupId]]
    AllocatedStorage: '5'
    DBInstanceClass: db.t3.small
    Engine: MySQL
    MasterUsername: admin
    MasterUserPassword: password
    DBSubnetGroupName:
      Ref: MyDBSubnetGroup
    DeletionPolicy: Snapshot

Outputs:
  VPC:
    Description: VPC Creada
    Value: !Join [ "", [ !Ref VPC, !FindInMap [ Variables, ProjectName, Value ] ] ]

```

```
PublicSubnetsA:
  Description: Subnet publica A creada
  Value: !Join [ "", [ !Ref PublicSubnetA, !FindInMap [ Variables, PublicSubnetRangeA,
Value ] ]]

PublicSubnetsB:
  Description: Subnet publica B creada
  Value: !Join [ "", [ !Ref PublicSubnetA, !FindInMap [ Variables, PublicSubnetRangeB,
Value ] ]]

PrivateSubnetsA:
  Description: Subnet privada A creada
  Value: !Join [ "", [ !Ref PrivateSubnetA, !FindInMap [ Variables, PrivateSubnetRangeA,
Value ] ]]

PrivateSubnetsB:
  Description: Subnet privada B creada
  Value: !Join [ "", [ !Ref PrivateSubnetB, !FindInMap [ Variables, PrivateSubnetRangeB,
Value ] ]]

NoIngressSecurityGroup:
  Description: Grupo de seguridad para VPC creado
  Value: !Ref NoIngressSecurityGroup

EC2SecurityGroup:
  Description: Grupo de seguridad para EC2 creado
  Value: !Ref EC2SecurityGroup

EC2InstanceA:
  Description: Instancia EC2 A creada
  Value: !Ref EC2InstanceA

EC2InstanceB:
  Description: Subnet privada B creada
  Value: !Ref EC2InstanceB

ELBSecurityGroup:
  Description: Grupo de seguridad para balanceador de carga creado
  Value: !Ref ELBSecurityGroup

GrupoDestinoELB:
  Description: Grupo de destino para las intancias A y B creado
  Value: !Ref GrupoDestinoELB

LoadBalancer:
  Description: Balanceador de carga creado
  Value: !Ref LoadBalancer

ListenerELB:
  Description: Listener para el balanceador de carga creado
  Value: !Ref ListenerELB

BBDDSecurityGroup:
  Description: Grupo de seguridad para la base de datos RDS creado
  Value: !Ref BBDDSecurityGroup

MyDB:
  Description: Base de datos RDS MySQL creada
  Value: !Ref MyDB
```