

Creación de la VPC y de las 2 subredes

En primer lugar, vamos a crear una nueva VPC utilizando Localstack (también podríamos reutilizar la del ejercicio 1 si quisiésemos). De la configuración nos vamos a guardar el vpcId para utilizarlo más adelante.

```
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 create-vpc --cidr-block 192.168.0.0/16
{
  "Vpc": {
    "OwnerId": "000000000000",
    "InstanceTenancy": "default",
    "Ipv6CidrBlockAssociationSet": [],
    "CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-8b3427862f66af50f",
        "CidrBlock": "192.168.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ],
    "Tags": [],
    "VpcId": "vpc-af2b9f170ed9fedd8",
    "State": "pending",
    "CidrBlock": "192.168.0.0/16",
    "DhcpOptionsId": "default"
  }
}
```

Ahora utilizaremos el siguiente comando para crear dos subredes (192.168.1.0/24 y 192.168.2.0/24) dentro de la VPC anteriormente creada.

```
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 create-subnet --vpc-id vpc-af2b9f170ed9fedd8 --cidr-block 192.168.1.0/24 --availability-zone us-east-1a
{
  "Subnet": {
    "AvailabilityZoneId": "us-east-1a",
    "OwnerId": "000000000000",
    "AssignIpv6AddressesOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [],
    "Tags": [],
    "SubnetArn": "arn:aws:ec2:us-east-1:000000000000:subnet/subnet-6d41ac8f15ca48a7",
    "Ipv6Native": false,
    "SubnetId": "subnet-6d41ac8f15ca48a7",
    "State": "pending",
    "VpcId": "vpc-af2b9f170ed9fedd8",
    "CidrBlock": "192.168.1.0/24",
    "AvailableIpAddressCount": 251,
    "AvailabilityZone": "us-east-1a",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false
  }
}
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 create-subnet --vpc-id vpc-af2b9f170ed9fedd8 --cidr-block 192.168.2.0/24 --availability-zone us-east-1a
{
  "Subnet": {
    "AvailabilityZoneId": "us-east-1a",
    "OwnerId": "000000000000",
    "AssignIpv6AddressesOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [],
    "Tags": [],
    "SubnetArn": "arn:aws:ec2:us-east-1:000000000000:subnet/subnet-7acac1fae2b99fe40",
    "Ipv6Native": false,
    "SubnetId": "subnet-7acac1fae2b99fe40",
    "State": "pending",
    "VpcId": "vpc-af2b9f170ed9fedd8",
    "CidrBlock": "192.168.2.0/24",
    "AvailableIpAddressCount": 251,
    "AvailabilityZone": "us-east-1a",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false
  }
}
```

Creación de la subred pública

Crear un Internet Gateway

Un Internet Gateway (IGW) permite la comunicación entre los recursos dentro de la VPC y la red pública (internet). Para hacer pública nuestra subred, necesitamos crear e integrar este componente.

Una vez creado el Internet Gateway, debemos vincularlo a nuestra VPC para habilitar la conectividad hacia y desde internet utilizando el InternetGatewayID

```
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 create-internet-gateway \
{
  "InternetGateway": {
    "Attachments": [],
    "InternetGatewayId": "igw-b8eb8ec1dfa36b121",
    "OwnerId": "000000000000",
    "Tags": []
  }
}
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 attach-internet-gateway \
--internet-gateway-id igw-0123456789abcde9 \
--vpc-id vpc-af2b9f170ed9fe6d8
An error occurred (InvalidInternetGatewayID.NotFound) when calling the AttachInternetGateway operation: InternetGatewayID igw-0123456789abcde9 does not exist.
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 attach-internet-gateway --internet-gateway-id igw-b8eb8ec1dfa36b121 --vpc-id vpc-af2b9f170ed9fe6d8
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack#
```

Crear y configurar una tabla de ruteo y asociarla a nuestra subred

Las tablas de ruteo determinan cómo se enruta el tráfico dentro de la red. En este paso ejecutaremos tres comandos:

- Creamos una tabla de ruteo personalizada que incluya una ruta hacia internet a través del Internet Gateway.
- Luego, agregamos una ruta a internet (toda dirección externa).
- Por último, conectamos la tabla de ruteo configurada con la subred para que esta pueda enrutar tráfico externo.

```
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 create-route-table \
--vpc-id vpc-af2b9f170ed9fe6d8
{
  "RouteTable": {
    "Associations": [],
    "RouteTableId": "rtb-88d883090e640aac7",
    "Routes": [
      {
        "DestinationCidrBlock": "192.168.0.0/16",
        "GatewayId": "local",
        "State": "active"
      }
    ],
    "Tags": [],
    "VpcId": "vpc-af2b9f170ed9fe6d8",
    "OwnerId": "000000000000"
  }
}
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 create-route \
--route-table-id rtb-88d883090e640aac7 \
--destination-cidr-block 0.0.0.0/0 \
--gateway-id igw-b8eb8ec1dfa36b121
{
  "Return": true
}
root@daniel-MS-7E28:/home/daniel/Sprint1/LocalStack# aws --profile localstack --endpoint-url=http://localhost:4566 ec2 associate-route-table \
--subnet-id subnet-6d41acb8f15ca48a7 \
--route-table-id rtb-88d883090e640aac7
{
  "AssociationId": "rtbassoc-8c8f925294755c49d"
}
```

Con esto, la subred queda correctamente configurada como pública.

Creación de la subred privada

La subred privada no necesita ninguna configuración extra. Simplemente no debe estar asociada a una tabla de ruteo con ruta a internet.