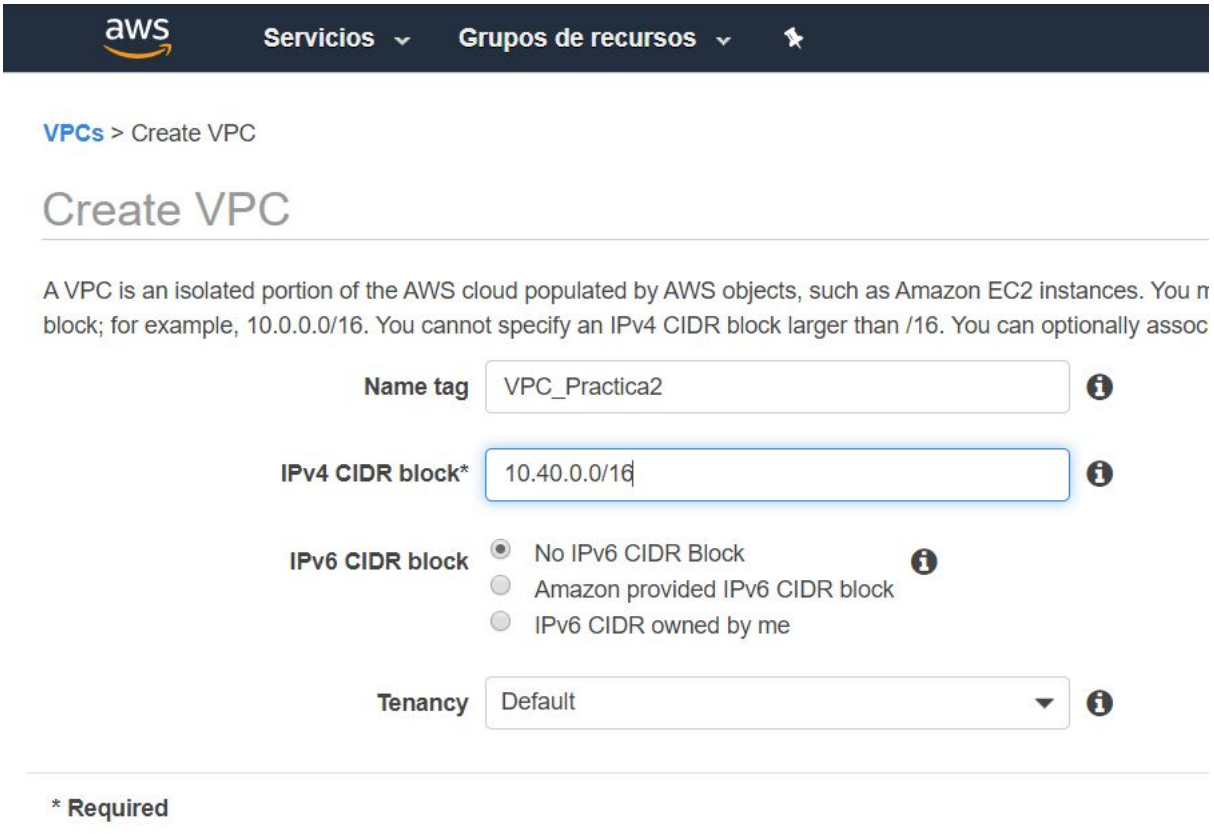
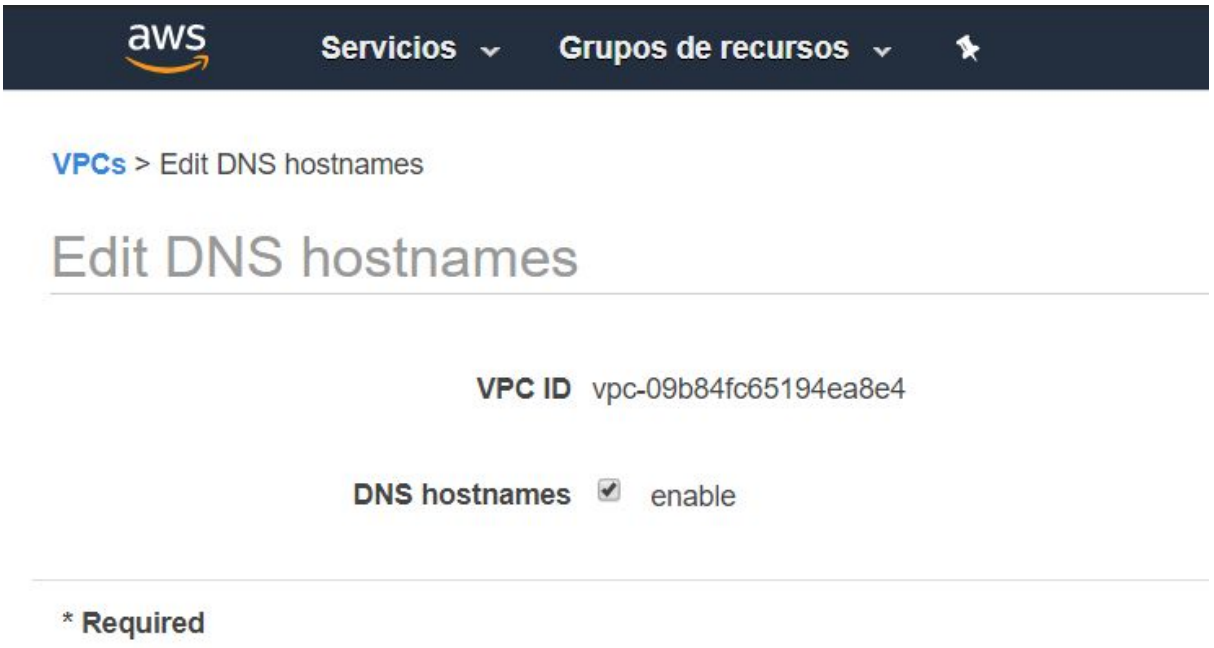
**GENERANDO LA ESTRUCTURA DE RED**

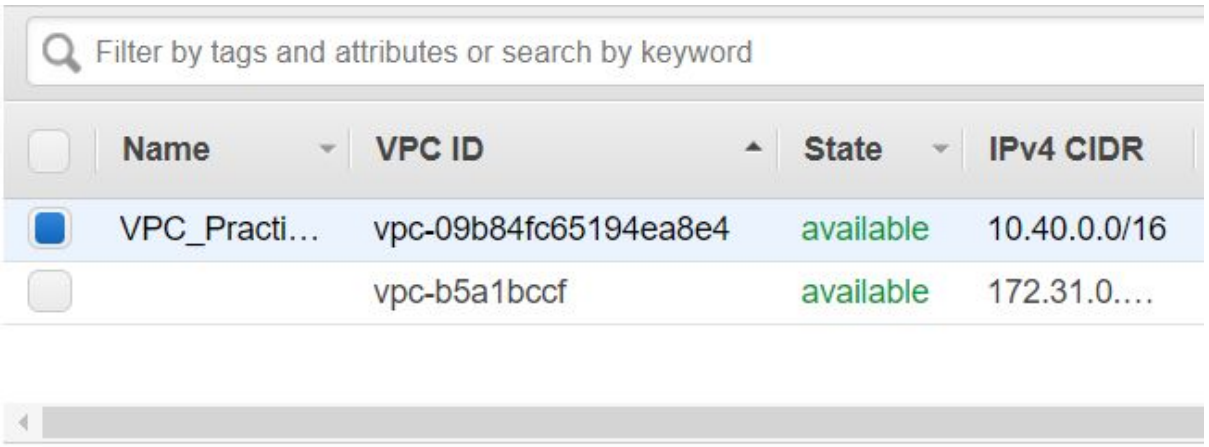
1º Creo la VPC con rango de direcciones 10.40.0.0/16




2º Habilito los nombres de dominio DNS



Características de  VPC_Practica2

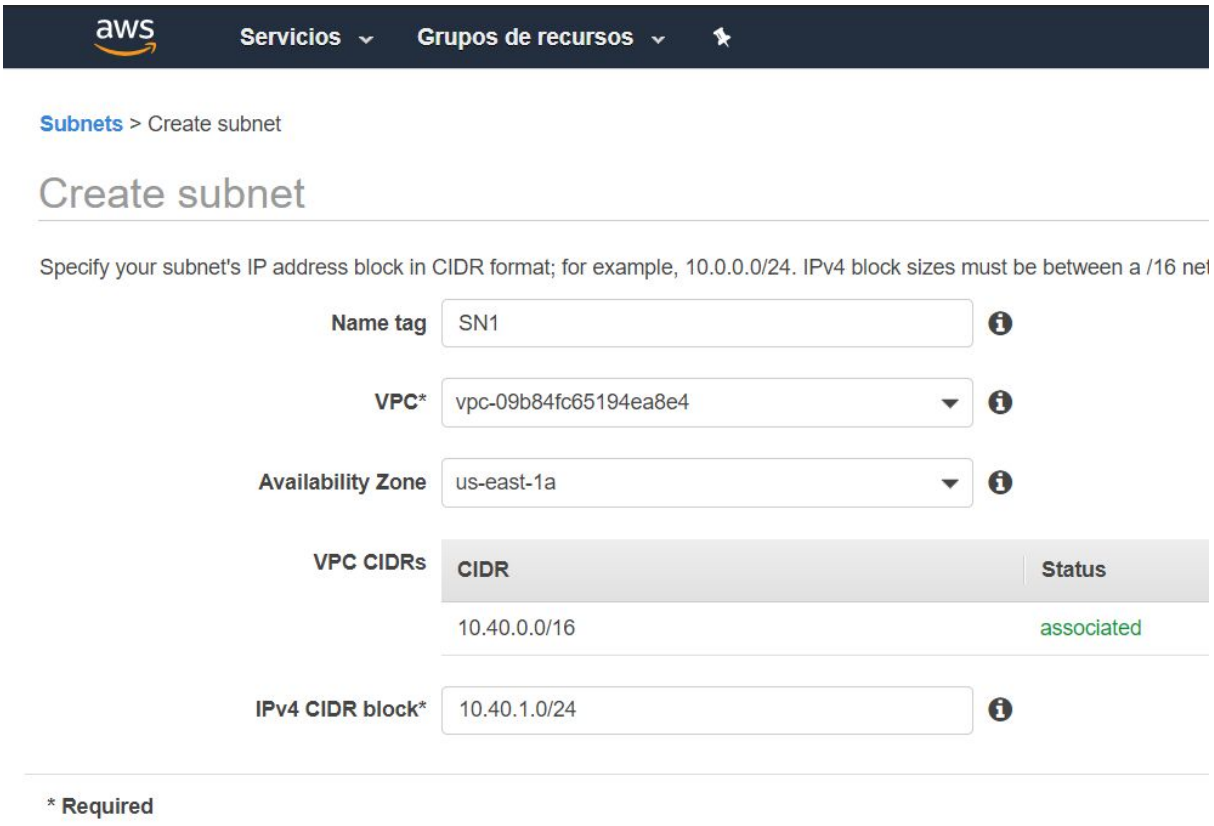


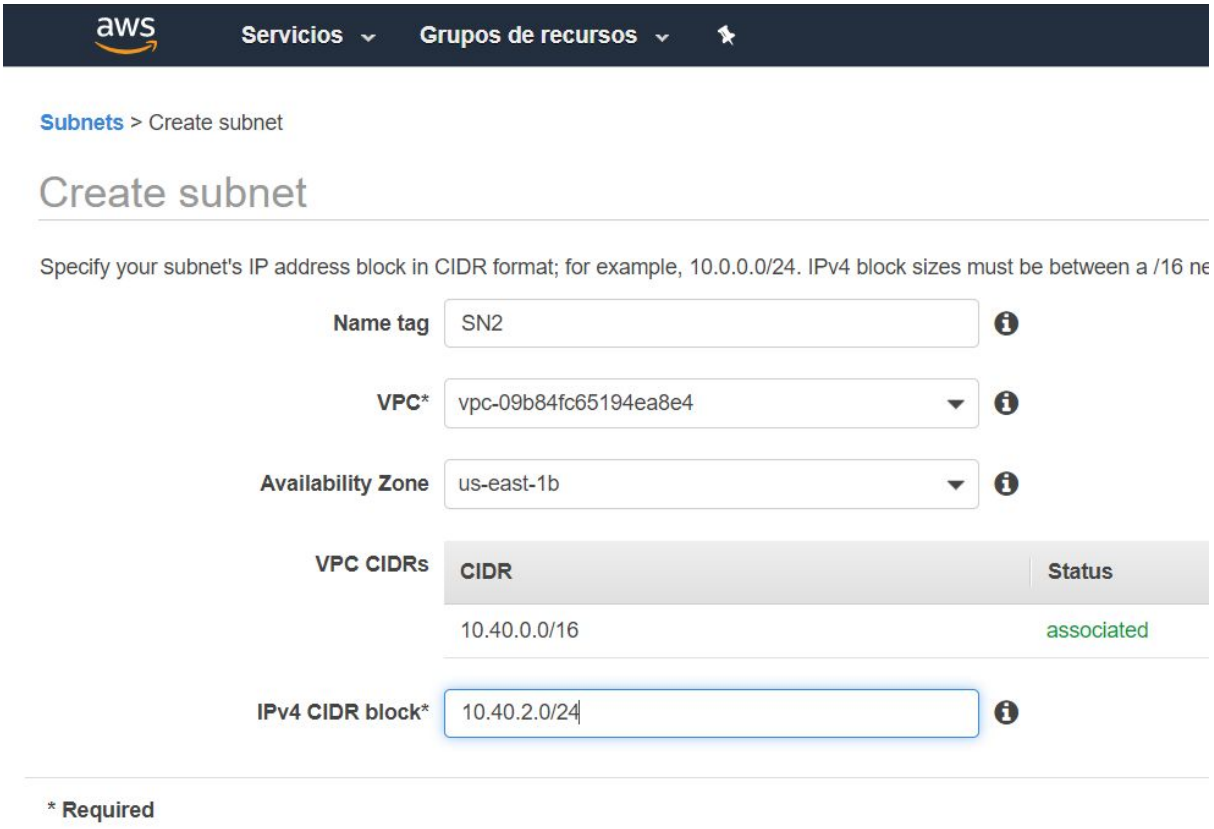

**VPC:** vpc-09b84fc65194ea8e4

**Description** | CIDR Blocks | Flow Logs | Tags

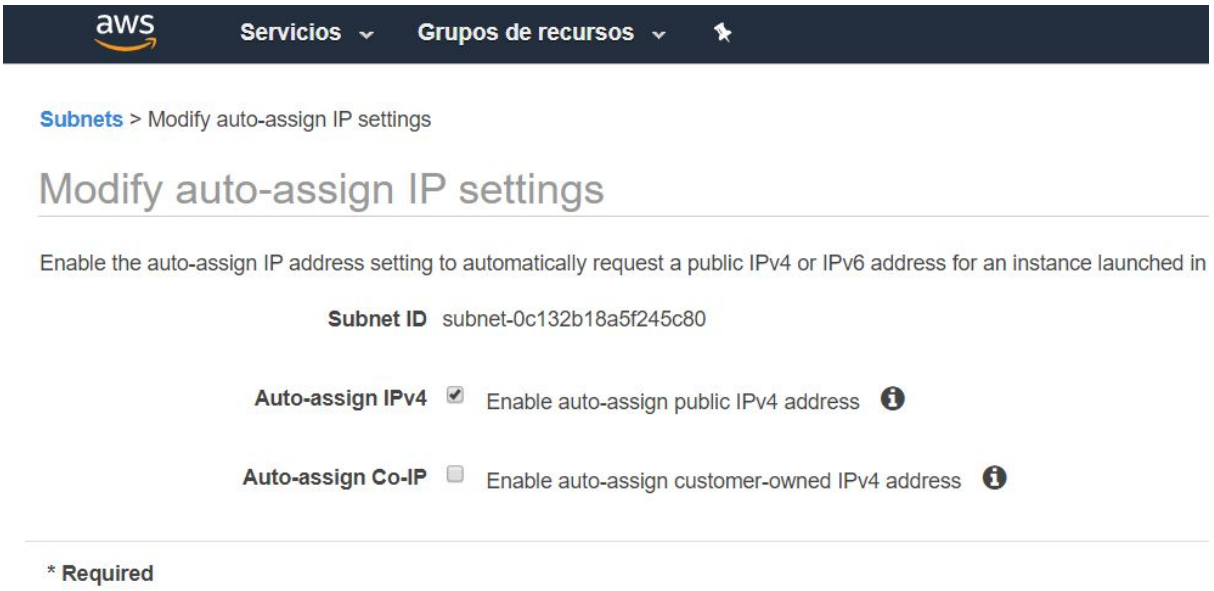
| | |
|---|---|
| **VPC ID** | vpc-09b84fc65194ea8e4 |
| **State** | available |
| **IPv4 CIDR** | 10.40.0.0/16 |
| **IPv6 CIDR** | - |
| **DNS resolution** | Enabled |
| **DNS hostnames** | Enabled |
| **ClassicLink DNS Support** | Disabled |
| **Owner** | 358816195022 |

3º Genero las dos Subnets (SN1=10.40.1.0/24 y SN2=10.40.2.0/24) en VPC_Practica2, cada una en una availability zone diferente.

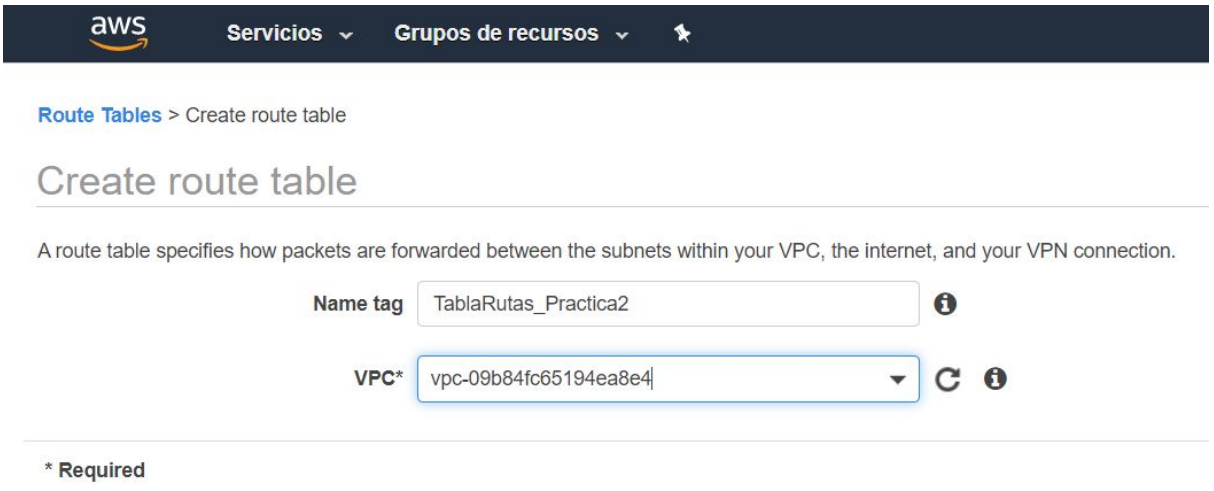


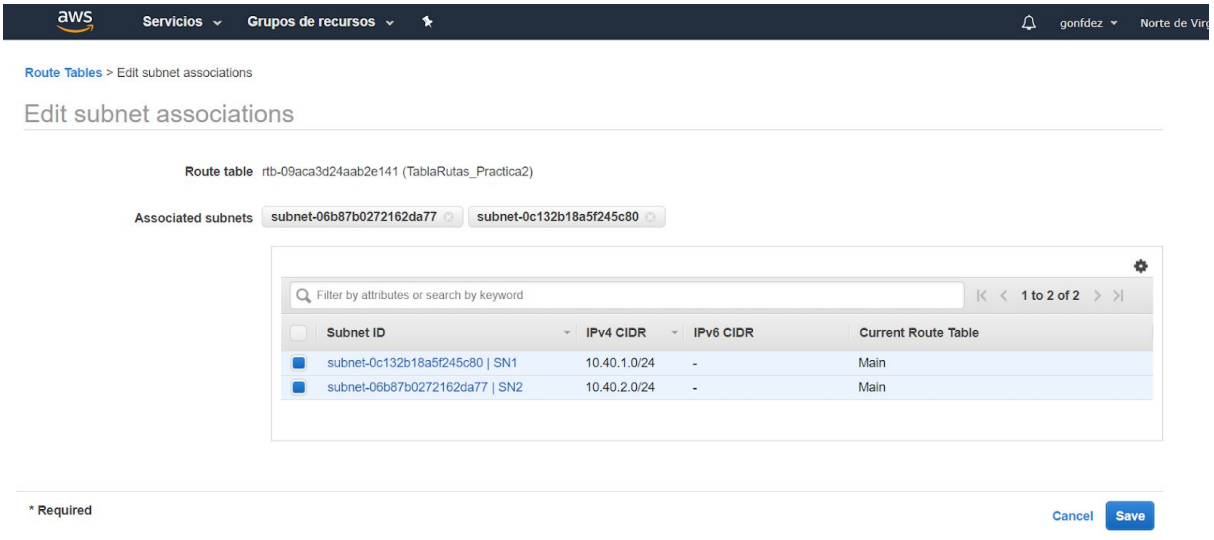
Configuro la Subnet SN1 (Que será la que tenga el servidor WEB) para que tenga direccionamiento público IPv4




4º Generando la tabla de rutas




Asocio las subnets SN1 y SN2 a la tabla de rutas

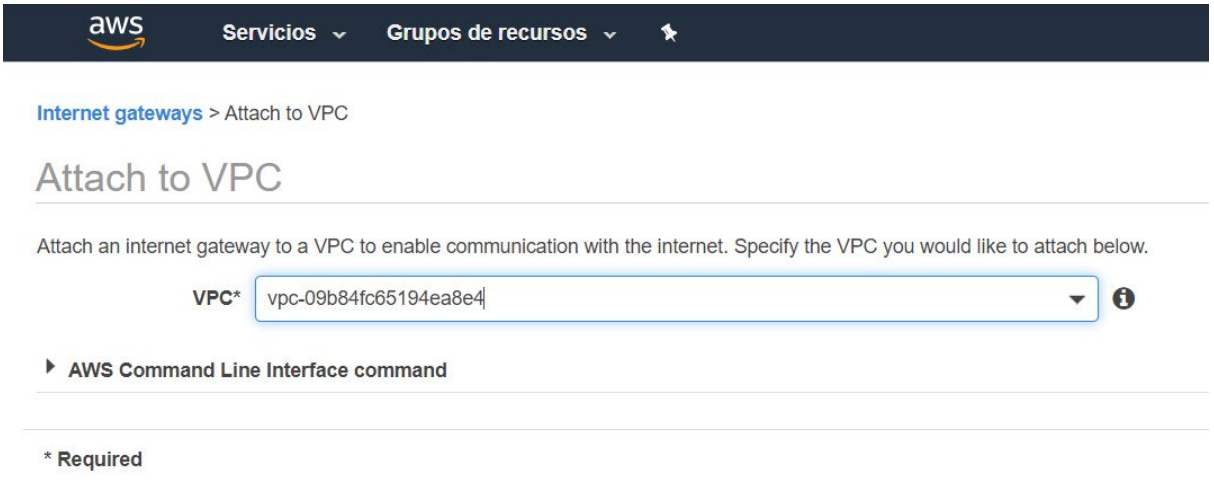



Ruta

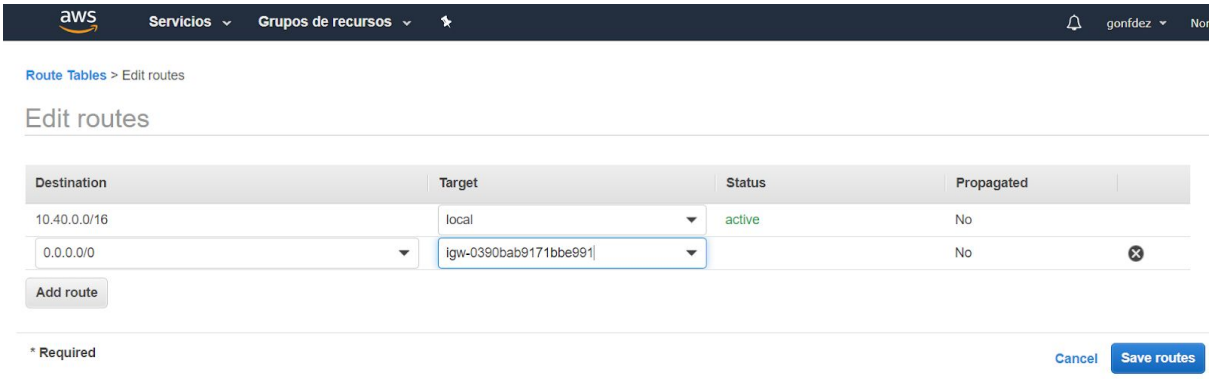
| Destination | Target | Status | Propagated |
|---|---|---|---|
| 10.40.0.0/16 | local | active | No |

5º Creo la Internet Gateway


Selecciono que dé internet a mi VPC_Practica2




Ahora me voy a la Tabla de rutas y configuro el router (la tabla de rutas) para que me encamine el tráfico a este gateway




En este punto ya tengo esto:

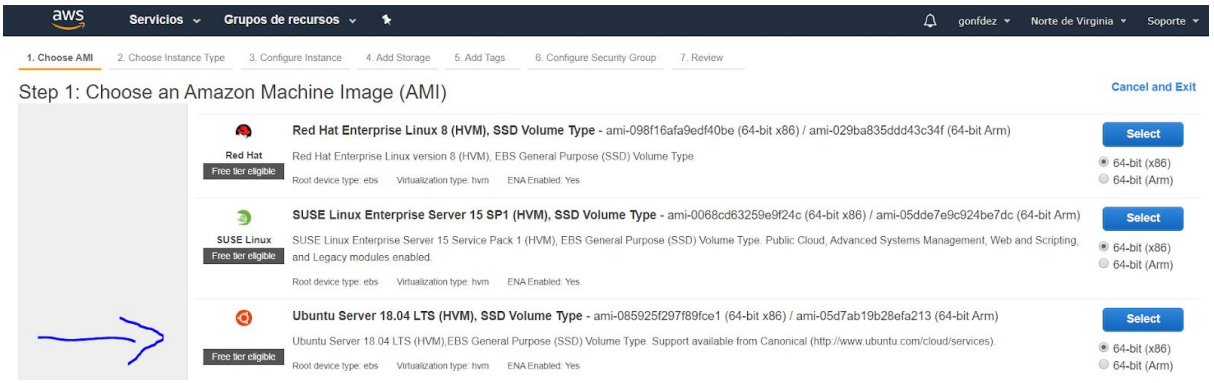
VPC_Practica2 GateWay

| router |
|---|

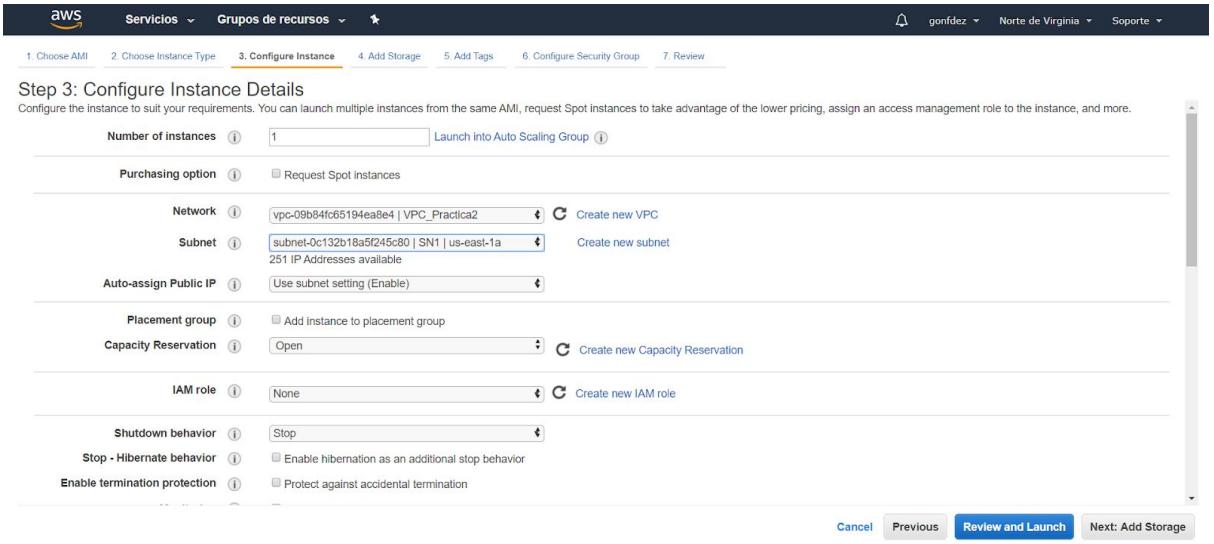
| | |
|---|---|
| SN1 (Vacio) | SN2 (Vacio) |

## GENERANDO LA MÁQUINA EC2 DE LA SN1

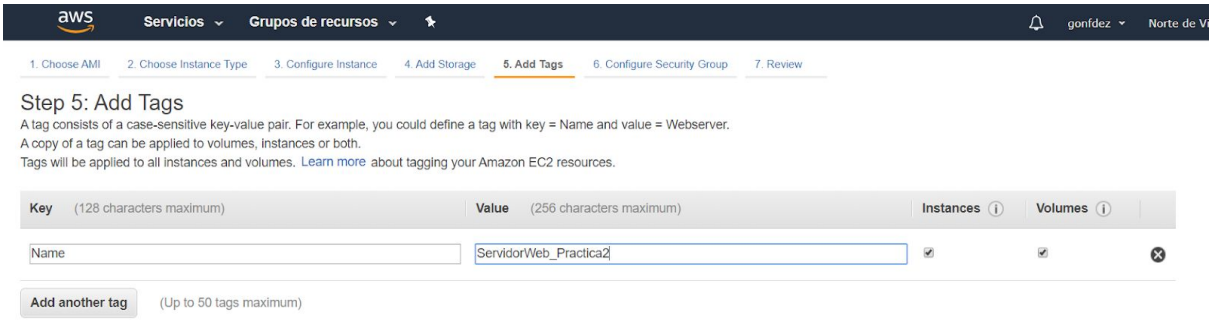
1º Elijo la imagen que quiero usar para la máquina virtual



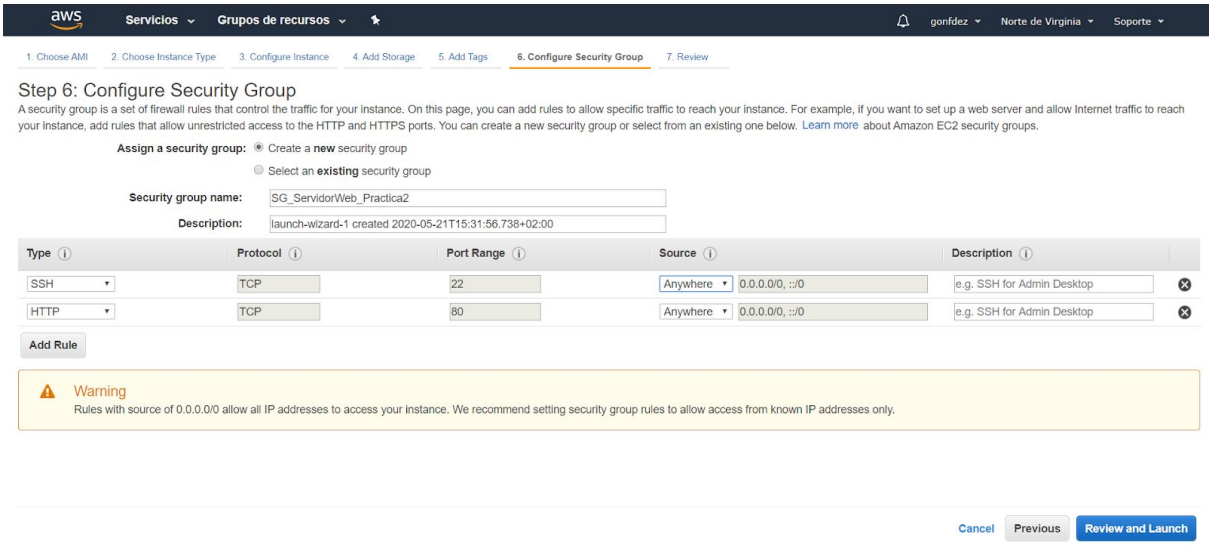

La asigno al VPC_Practica2 y a la Subnet pública SN1



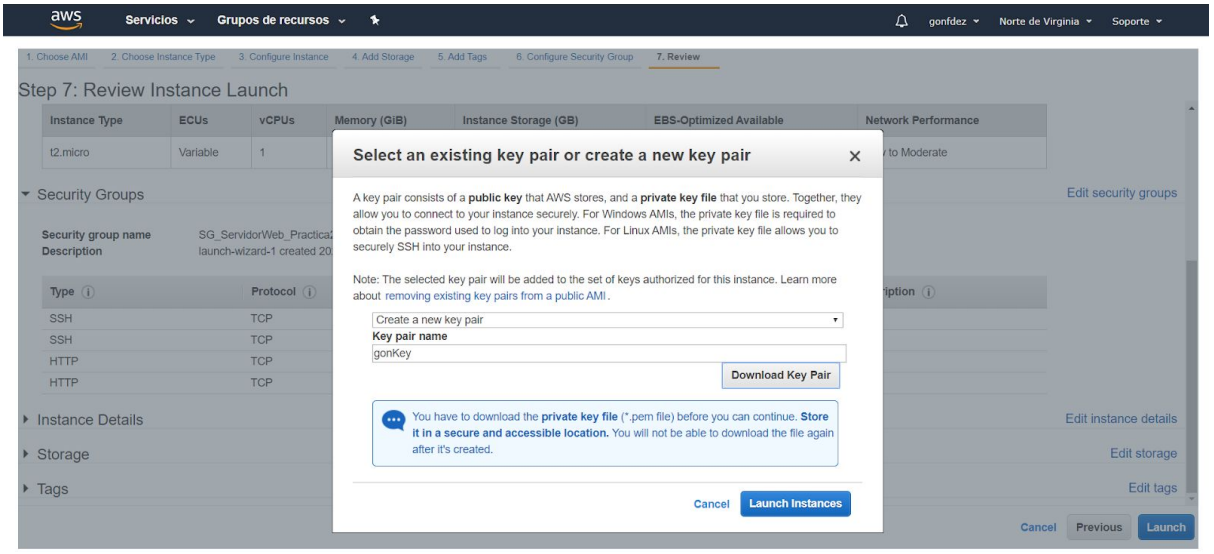

Pongo una etiqueta a esta máquina “ServidorWeb_Practica2”



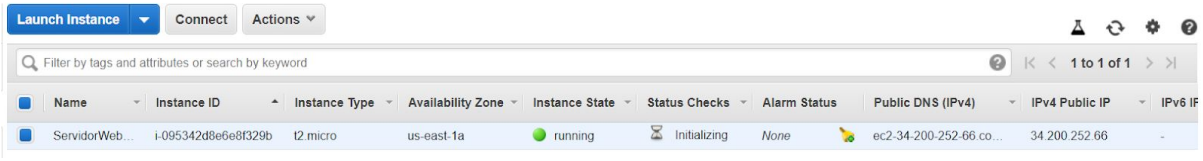
Le asigno un security group “SG_ServidorWeb_Practica2” nuevo que solo permite servicios SSH y HTTP




Lo lanzo y le asigno un fichero de claves “gonKey”

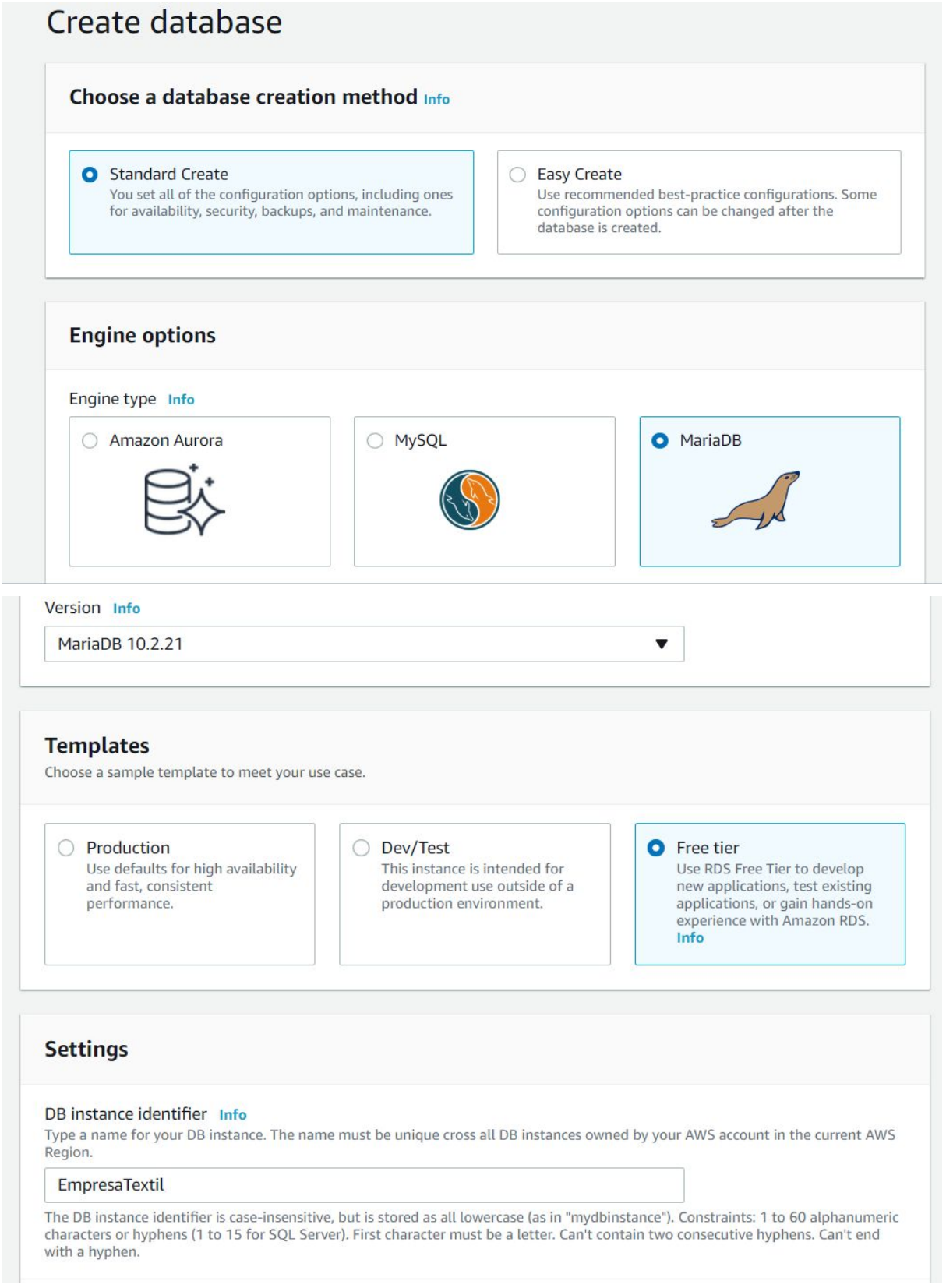



Listo



**GENERANDO LA MÁQUINA RDS de MARIADB DE LA SN2**

1º Creo la database



▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter

☐ Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password Info

••••••••••••

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password Info

••••••••••••

## DB instance size

DB instance class Info
Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

○ Standard classes (includes m classes)
○ Memory Optimized classes (includes r and x classes)
◉ Burstable classes (includes t classes)

db.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized ▼

○ Include previous generation classes

## Storage

Storage type Info

General Purpose (SSD) ▼

Allocated storage

20 GiB

(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage **may improve** IOPS performance.

### Storage autoscaling Info
Provides dynamic scaling support for your database's storage based on your application's needs.

☑ Enable storage autoscaling
Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

Maximum storage threshold Info
Charges will apply when your database autoscales to the specified threshold

1000 GiB

Minimum: 21 GiB, Maximum: 16384 GiB

## Availability & durability

Multi-AZ deployment Info

○ Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

○ Do not create a standby instance

## Connectivity

Virtual private cloud (VPC) Info
VPC that defines the virtual networking environment for this DB instance.

VPC_Practica2 (vpc-09b84fc65194ea8e4)

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group Info
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

Create new DB Subnet Group

Publicly accessible Info

- Yes
  Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.
- No
  RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

VPC security group
Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

- Choose existing
  Choose existing VPC security groups
- Create new
  Create new VPC security group

New VPC security group name

SG_RDS_Practica2

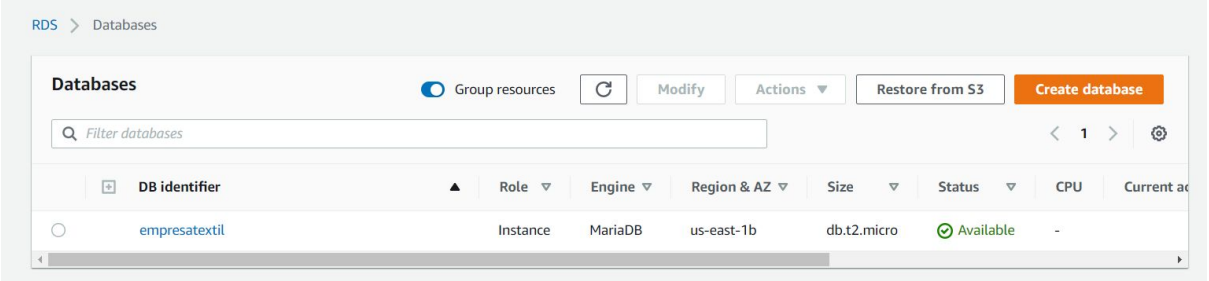
Availability Zone Info

us-east-1c

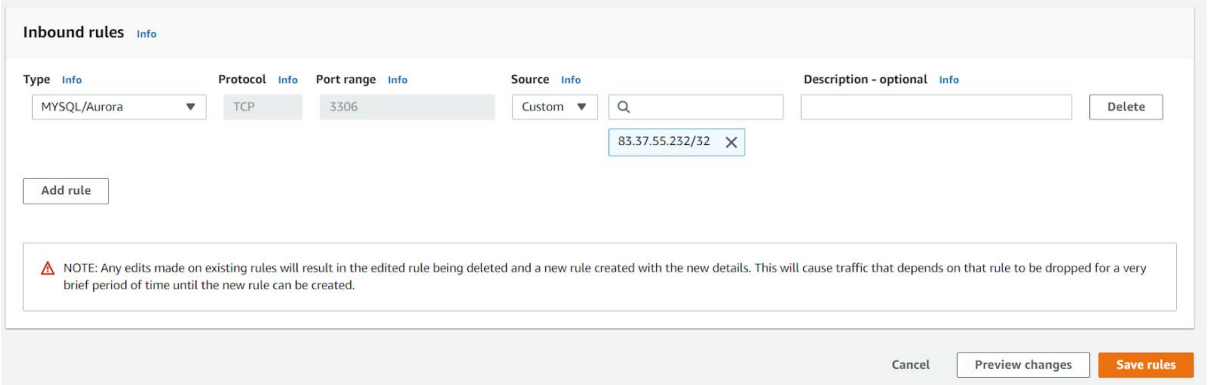
Database port Info
TCP/IP port that the database will use for application connections.

3306

Listo



2º Configuro el security group para que solo permita servicio Mysql por el puerto 3306




## INSTALACIÓN DEL SERVIDOR WEB APACHE Y HTTP SOBRE LA EC2

1º Me conecto a mi instacia EC2 desde mi máquina Debian10

```
[dev] gonfs@debian:~/practica2
$ chmod 400 gonKey.pem
[dev] gonfs@debian:~/practica2
$ ssh -i "gonKey.pem" ubuntu@ec2-34-200-252-66.compute-1.amazonaws.com
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu May 21 14:45:40 UTC 2020

  System load:  0.08              Processes:           86
  Usage of /:   13.7% of 7.69GB   Users logged in:     0
  Memory usage: 15%               IP address for eth0: 10.40.1.91
  Swap usage:   0%

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-40-1-91:~$
```

2ºInstalo el servidor web apache

```
ubuntu@ip-10-40-1-91:~$ sudo apt-get -y install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
```

…

```
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib
/systemd/system/apache-htcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.39) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
ubuntu@ip-10-40-1-91:~$
```

```
ubuntu@ip-10-40-1-91:~$ ps -ef | grep apache
root      2174     1  0 14:47 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2176  2174  0 14:47 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  2177  2174  0 14:47 ?        00:00:00 /usr/sbin/apache2 -k start
ubuntu   13524  1639  0 14:49 pts/0    00:00:00 grep --color=auto apache
ubuntu@ip-10-40-1-91:~$
```

```
ubuntu@ip-10-40-1-91:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Thu 2020-05-21 14:47:44 UTC; 2min 28s ago
 Main PID: 2174 (apache2)
    Tasks: 55 (limit: 1152)
   CGroup: /system.slice/apache2.service
           ├─2174 /usr/sbin/apache2 -k start
           ├─2176 /usr/sbin/apache2 -k start
           └─2177 /usr/sbin/apache2 -k start

May 21 14:47:44 ip-10-40-1-91 systemd[1]: Starting The Apache HTTP Server...
May 21 14:47:44 ip-10-40-1-91 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-10-40-1-91:~$
```

```
ubuntu@ip-10-40-1-91:~$ sudo apt-get install php libapache2-mod-php php-mysql
```

Copio el archivo Conexion_Mysql.php a mi maquina EC2 en el directorio /var/www/html/practica2

```
[dev] gonfs@debian:~/practica2
$ scp -i "gonKey.pem" Conexion_MySQL.php ubuntu@ec2-34-200-252-66.compute-1.amazonaws.com:/var
/www/html/practica2
Conexion_MySQL.php                                    100% 2116    22.6KB/s   00:00
```

Copio el contenido de Conexion_Mysql.php en index.html

```
ubuntu@ip-10-40-1-91:/var/www/html$ sudo mv practica2/Conexion_MySQL.php index.html
```

**CONEXIÓN A LA RDS DESDE LA EC2**

```
ubuntu@ip-10-40-1-91:~$ sudo apt-get -y update
```

```
ubuntu@ip-10-40-1-91:~$ sudo apt-get install mariadb-client
```

me conecto a mi RDS

```
ubuntu@ip-10-40-1-91:~$ mariadb -h empresatextil.crovm6ynd5nh.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
```

Se queda así la terminal, no me conecta y me dice que ha pasado el tiempo de conexion y no responde.