

# PRACTICA AWS EC2

- 1) Creamos una Aplicación que use una BBDD local MySQL de coches y la subimos a un repositorio en GitHub

## EC2 DEDICADA BBDD MARIADB

- 2) En la consola AWS creamos una instancia EC2 en la zona **us-east-1d**.  
La agregamos a un Security Group con los **puertos MySql/Aurora 3306,HTTP 80 y SSH 22 habilitados a cualquiera** (de momento)

Nos conectamos a la EC2

```
sudo su
```

- 3) Instalamos MariaDB

```
sudo yum install mariadb-server  
service mariadb start
```

- 4) Configuramos el acceso root

```
mysql_secure_installation
```

como root no tiene password (nos pide password de root, lo dejamos vacío) nos pedirá uno nuevo.

Hacemos que el server arranque al inicio maria\_db

```
chkconfig mysqld on
```

Entramos a mysql

```
mysql -u root -p
```

- 5) Damos permiso para acceder remotamente como root (no es recomendable, lo suyo sería hacer un usuario aparte)

```
GRANT ALL PRIVILEGES ON *.* to root@localhost WITH GRANT OPTION;
```

```
GRANT ALL PRIVILEGES ON *.* to root@'%' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

En caso de querer crear un usuario remoto con contraseña:

```
CREATE USER 'usuarioremoto'@'localhost' IDENTIFIED BY 'password';
```

```
CREATE USER 'usuarioremoto'@'%' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON *.* to usuarioremoto@localhost WITH GRANT OPTION;
```

```
GRANT ALL PRIVILEGES ON *.* to usuarioremoto@'%' WITH GRANT OPTION;
```

- 6) Modificamos los ficheros de configuración de MariaDb

```
cd /etc
```

```
nano my.cnf
```

Editamos debajo de [mysqld] añadiendo

```
port=3306
```

```
bind-address=0.0.0.0
```

(Por ahora, acceso desde cualquier IP)

Reiniciamos el servicio

```
Service mariadb restart
```

Comprobamos desde nuestro equipo (mediante Workbench por ejemplo) que podemos conectarnos a la base de datos usando como cadena de conexión la **IP publica de nuestra EC2** y usuario **root** (sin password). Ahora podemos crear las tablas para la Aplicación Core.

## EC2 DEDICADA A APLICACIÓN CORE

- 1) Vamos a la consola AWS y creamos una nueva EC2 dedicada para alojar la Aplicación Core en la misma región que la EC2 de la BBDD (ponemos la misma región, pero puede ser distinta zona de disponibilidad, en este caso, **us-east-1a**)

La añadiremos al mismo grupo de seguridad que la EC2 de MariaDB y accedemos.

- 2) Instalamos **Apache, GitHub, Dotnet y libunwind**

```
yum update -y
yum install httpd -y
yum install git -y
sudo rpm -Uvh https://packages.microsoft.com/config/centos/7/packages-microsoft-prod.rpm
sudo yum install dotnet-sdk-3.1 -y
sudo yum install dotnet-runtime-5.0 -y
sudo yum install libunwind -y
systemctl start httpd
systemctl enable httpd
```

- 3) Hacemos **Table Reverse Routing** para que al acceder a través del puerto 80 de nuestro Apache, la petición se redirija al kestrel de Core (donde alojaremos la Aplicación Core)

```
sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 5000
```

- 4) Accedemos a git para descargar el repositorio del proyecto en una carpeta dentro de **/var** y con dotnet reconstruimos el proyecto con sus respectivos NuGet

**\*\*Previamente debemos modificar en el proyecto la cadena de conexión de la BBDD para que apunte a la ip privada del EC2 MariaDB con usuario root SIN PASSWORD (ya que root@'%' no tiene asignada ninguna contraseña) debido a que estarán en la misma región us-east, pero en distintas zonas de acceso 1a y 1d**

```
git config --global user.name "usuariogit"
git config --global user.email "email de git"
```

```
cd /var
mkdir coreappcoches
cd coreappcoches
git clone urldelrepo
```

Damos permisos para compilar

```
sudo chown -R ec2-user /var/coreappcoches
```

Vamos donde esté situado el **.csproj** del proyecto y compilamos con dotnet

```
dotnet restore
dotnet build
```

Lanzamos la app en el puerto 5000 y entramos a probarla a través de la IP publica en cualquier navegador

```
dotnet run --urls "http://0.0.0.0:5000"
```

**\*\* Aunque estén ambas EC2 interactuando perfectamente entre BBDD y APP, lo normal sería ahora ir al grupo de seguridad en AWS y cambiar el acceso al puerto 3306 ÚNICAMENTE a su grupo de seguridad, de forma que SOLAMENTE la EC2 -APP tenga acceso a la BBDD**