



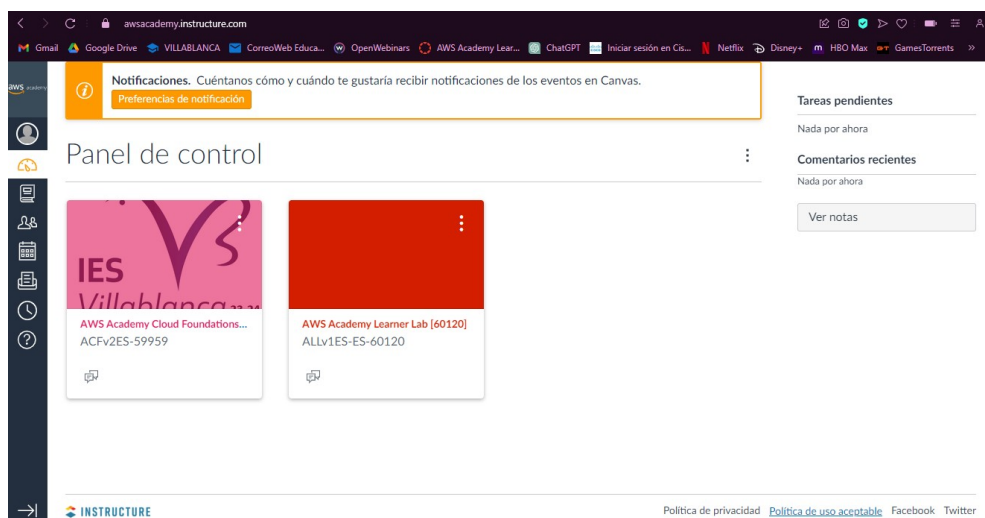
CREAR UN SERVIDOR DE BASE DE DATOS AWS

Andrea Díaz Maeso
2º CFGS ASIR

ÍNDICE

1. REGISTRARSE EN AWS ACADEMY.....	1
2. CREAR EL SERVIDOR DEBIAN 12 EN EL APARTADO DE LEARNER LABS.....	1
3. CONECTARTE POR SSH AL SERVIDOR E INSTALAR MARIADB.....	10
4. CREAR UN USUARIO PERSONALIZADO CON TODOS LOS PERMISOS.....	15

1. REGISTRARSE EN AWS ACADEMY



2. CREAR EL SERVIDOR DEBIAN 12 EN EL APARTADO DE LEARNER LABS

Lo primero que tenemos que hacer es descargar la clave que nos permitirá conectarnos por SSH a la instancia EC2 que vamos a crear. Para ello, hacemos clic en “**AWS Details**” y seleccionamos la clave privada en formato **PPK** (compatible con Putty).

Cloud Access

AWS CLI: Show

Cloud Labs
Remaining session time: 03:41:57(222 minutes)
Session started at: 2023-10-20T03:09:19-0700
Session to end at: 2023-10-20T07:09:19-0700

Accumulated lab time: 00:17:00 (17 minutes)

No running instance

SSH key Show Download PEM **Download PPK**

AWS SSO Download URL

AWSAccountId	752207831009
Region	us-east-1

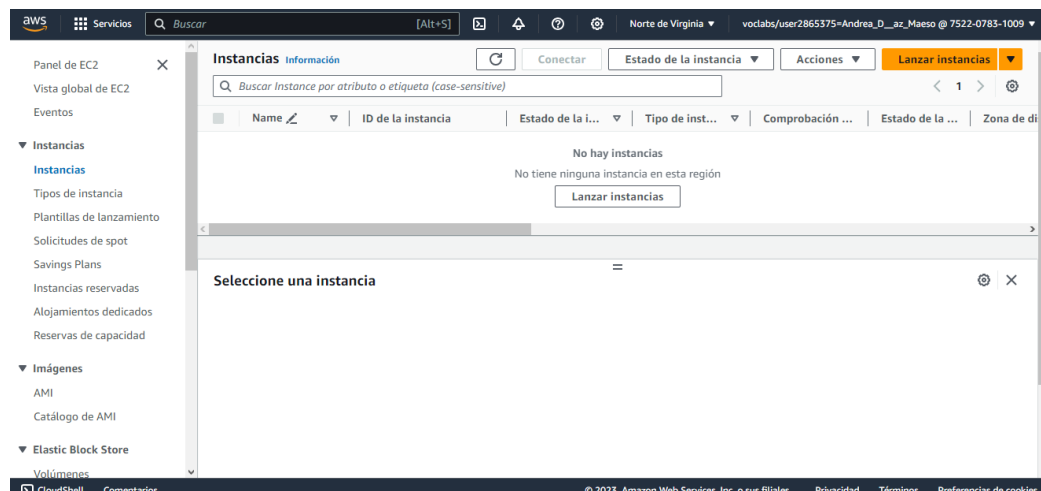
Con esto habremos descargado el archivo **labuser.ppk**.



Ahora, buscamos el servicio **“Amazon EC2”** desde el menú desplegable de **“Servicios”** (en la sección **“Informática”**) y abrimos la consola del servicio. Una vez allí, hacemos clic sobre **“Instancias”**:



Desde la siguiente ventana, hacemos clic en el botón **“Lanzar instancias”**:



A continuación, introducimos en el campo **“Nombre”** el valor **ServidorWeb** y seleccionamos en el apartado de **“Imágenes de aplicaciones y sistemas operativos (Amazon Machine Image)”** la AMI que contendrá la imagen del disco raíz que tendrá nuestra instancia. Elegimos la AMI de Amazon **Debian 12**:

Nombre y etiquetas [Información](#)

Nombre

ServidorWeb

[Agregar etiquetas adicionales](#)



debian

Debian

Apto para la capa gratuita

Proveedor verificado

Debian 12 (HVM), SSD Volume Type

ami-06db4d78cb1d3bbf9 (64 bits (x86)) / ami-0d3eda47adff3e44b (64 bits (Arm))

Debian 12 (HVM), EBS General Purpose (SSD) Volume Type. Community developed free GNU/Linux distribution. <https://www.debian.org/>

Plataforma: debian Tipo de dispositivo raíz: ebs Virtualización: hvm Habilitado para ENA: Sí

Seleccionar

☒ 64 bits (x86)

☐ 64 bits (Arm)

Ahora, debemos elegir el tipo de instancia. En nuestro caso, elegiremos un tipo de instancia de propósito general, **t2.micro**.

▼ Tipo de instancia [Información](#)

Tipo de instancia

t2.micro

Apto para la capa gratuita

Familia: t2 1 vCPU 1 GiB Memoria Generación actual: true

Bajo demanda Windows base precios: 0.0162 USD por hora

Bajo demanda SUSE base precios: 0.0116 USD por hora

Bajo demanda RHEL base precios: 0.0716 USD por hora

Bajo demanda Linux base precios: 0.0116 USD por hora

☒ Todas las generaciones

[Comparar tipos de instancias](#)

Se aplican costos adicionales a las AMI con software preinstalado

En el apartado “**Par de claves (inicio de sesión)**” seleccionamos la opción marcada como “**vockey**”. La clave pública permanecerá almacenada en la instancia EC2, mientras que nosotros custodiaremos la clave privada que descargamos anteriormente.

▼ Par de claves (inicio de sesión) [Información](#)

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

vockey



[Crear un nuevo par de claves](#)

En el apartado **“Configuraciones de red”**, se configuran la red y grupo de seguridad (cortafuegos) que se aplicarán a la instancia EC2. Para ello, presionamos el botón **“Editar”**, y seleccionamos desde el campo **“VPC”** el valor de la **VPC predeterminada**. En el campo **“Subred”**, elegimos la subred que se encuentre en la zona de disponibilidad **“us-east-1a”**. Verificaremos que el campo **“Asignar automáticamente IP pública”** se encuentra en el valor **“Habilitar”**:

▼ Configuraciones de red [Información](#)

VPC - *obligatorio* [Información](#)

vpc-086f1efddb690de4d (predeterminado) ▼ 

172.31.0.0/16



Subred [Información](#)

subnet-0027cf6c1ca7afe4b ▼ 

VPC: vpc-086f1efddb690de4d Propietario: 752207831009
Zona de disponibilidad: us-east-1a Direcciones IP disponibles: 4091
CIDR: 172.31.0.0/20

Asignar automáticamente la IP pública [Información](#)

Habilitar ▼

 [Crear nueva subred](#) 

A continuación, configuraremos el grupo de seguridad de la instancia EC2. Para ello, desde el apartado **“Firewall (grupos de seguridad)”**, seleccionaremos la opción **“Crear grupo de seguridad”**, asignamos como **“Nombre del grupo de seguridad”** el valor **“default-vpc-http-ssh”**, en la **“Descripción”** introduciremos **“Puertos HTTP y SSH”** y por último, definimos dos reglas de entrada para permitir el tráfico a nuestra instancia por los **puertos 80 TCP y 22 TCP**:

Firewall (security groups) [Información](#)

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☒ Crear grupo de seguridad

☐ Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad - *obligatorio*

default-vpc-http-ssh

Este grupo de seguridad se agregará a todas las interfaces de red. El nombre no se puede editar después de crear el grupo de seguridad. La longitud máxima es de 255 caracteres. Caracteres válidos: a-z, A-Z, 0-9, espacios y _-:/() #,@[]+= &; {}! \$*

Descripción - *obligatorio* [Información](#)

Puertos HTTP y SSH

Reglas de grupos de seguridad de entrada

▼ Regla del grupo de seguridad 1 (TCP, 22, 0.0.0.0/0)

Eliminar

Tipo [Información](#)

ssh ▼

Protocolo [Información](#)

TCP

Intervalo de puertos [Información](#)

22

Tipo de origen [Información](#)

Cualquier lugar ▼

Origen [Información](#)

🔍 Agregue CIDR, lista de prefijos

0.0.0.0/0 ✕

Descripción - optional [Información](#)

por ejemplo, SSH para Admin Desi

▼ Regla del grupo de seguridad 2 (TCP, 80, 0.0.0.0/0)

Eliminar

Tipo [Información](#)

HTTP ▼

Protocolo [Información](#)

TCP

Intervalo de puertos [Información](#)

80

Tipo de origen [Información](#)

Cualquier lugar ▼

Origen [Información](#)

🔍 Agregue CIDR, lista de prefijos

Descripción - optional [Información](#)

por ejemplo, SSH para Admin Desi

En el apartado de **“Configurar almacenamiento”**, especificamos el tamaño del volumen **EBS (Elastic Block Storage)** raíz, desde donde arrancará nuestra instancia EC2. En nuestro caso, seleccionamos un volumen de **8 GiB basado en gp3 (General Purpose de 3ª Generación)**:

▼ Configurar almacenamiento [Información](#)

[Avanzado](#)

1x

8

GiB

gp3 ▼

Volumen raíz (Sin cifrar)

Agregar un nuevo volumen

0 x sistemas de archivos

[Editar](#)

Por último, desde el cuadro **“Resumen”**, seleccionamos el **número de instancias (1)** y presionamos el botón **“Lanzar instancia”**:

▼ Resumen

Número de instancias [Información](#)

1

[Imagen de software \(AMI\)](#)

Debian 12 (HVM), SSD Volume Ty...[más](#)

[información](#)

ami-06db4d78cb1d3bbf9

[Tipo de servidor virtual \(tipo de instancia\)](#)

t2.micro

[Firewall \(grupo de seguridad\)](#)

Nuevo grupo de seguridad

[Almacenamiento \(volúmenes\)](#)

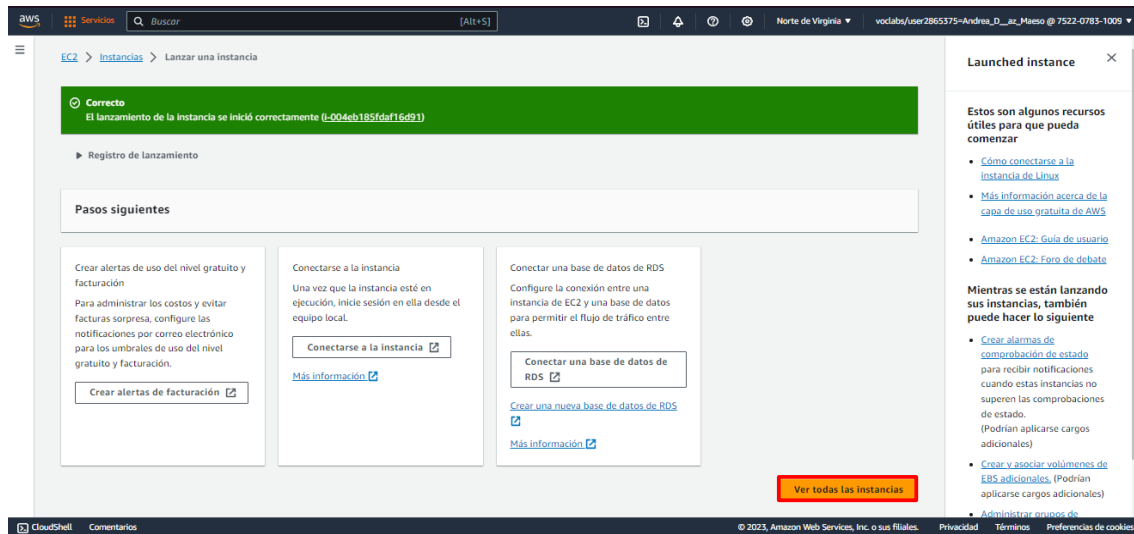
1 volumen(es): 8 GiB

Cancelar

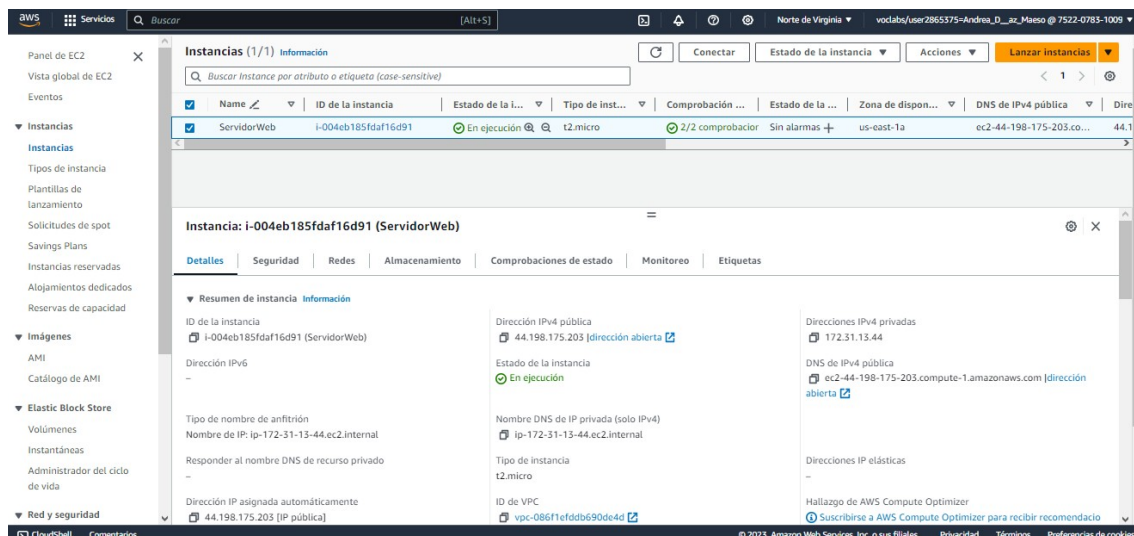
Lanzar instancia

[Revisar comandos](#)

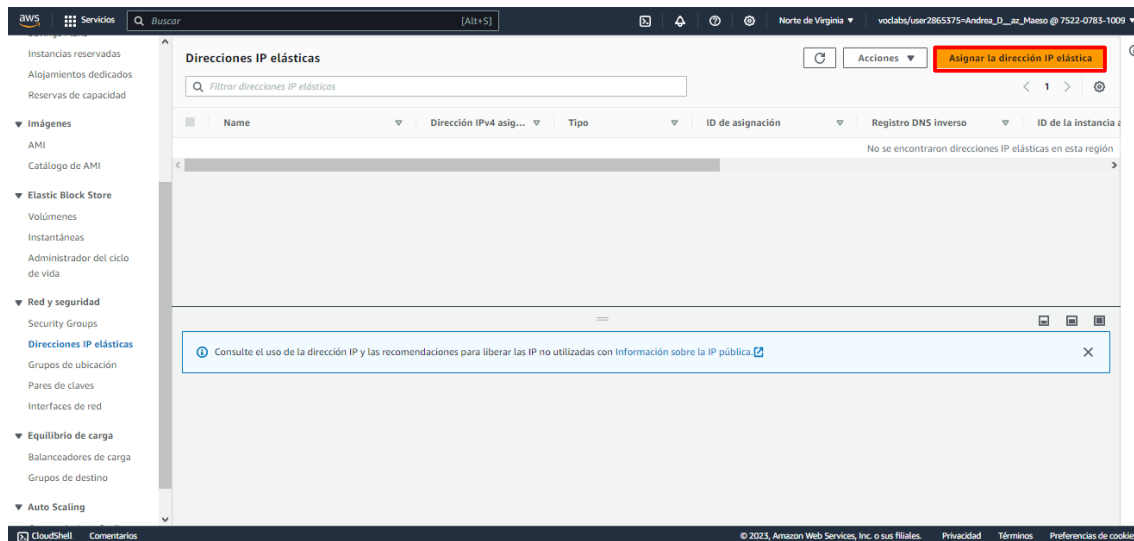
Con esto Amazon EC2 ha comenzado con el aprovisionamiento de nuestra instancia. Presionamos el botón “Ver todas las instancias”:



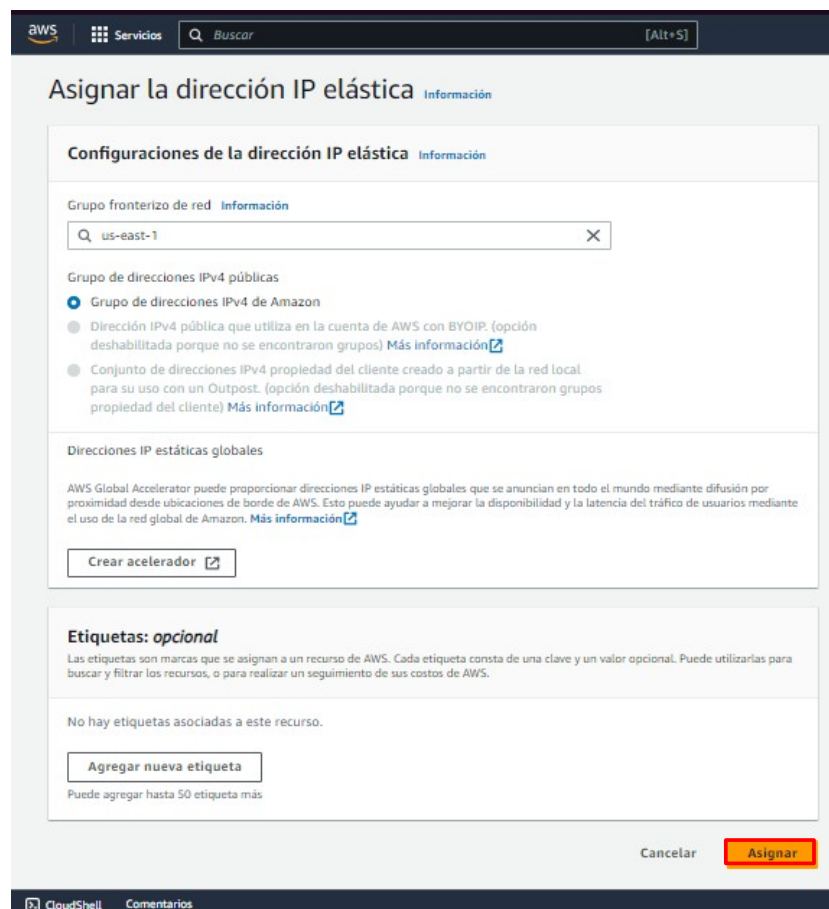
Podemos ver que nuestra instancia ya está operativa desde la consola. Si hacemos clic en el nombre de la instancia podemos ver sus detalles en la parte inferior de la ventana.



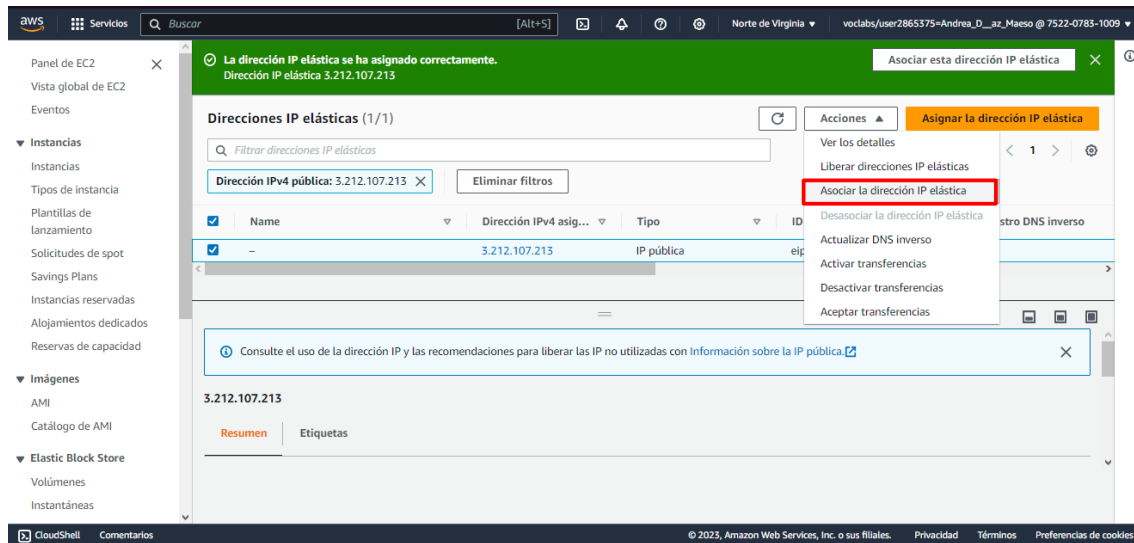
A continuación, tenemos que crear una **IP elástica (EIP)**. Para ello, nos dirigimos al apartado **“Red y Seguridad”** de la consola y seleccionamos la opción **“Direcciones IP elásticas”**. Desde allí, presionamos el botón **“Asignar la dirección IP elástica”**:



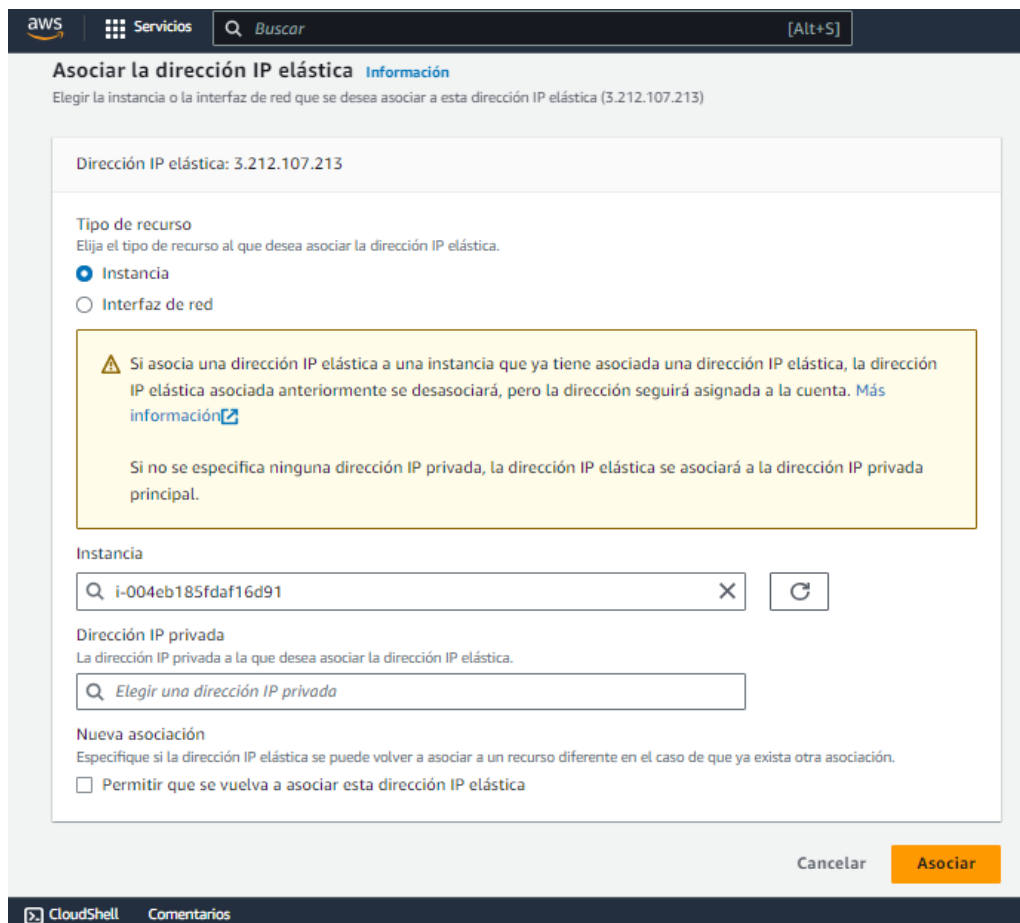
En la siguiente ventana, dejamos la configuración tal cual y presionamos el botón **“Asignar”**:



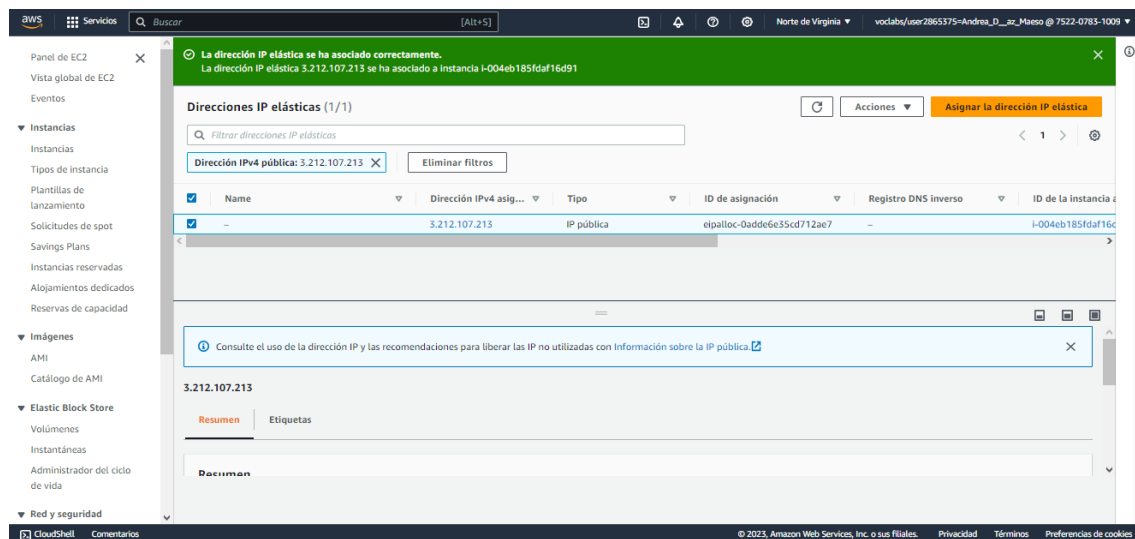
Ahora, asignamos la EIP a nuestra instancia seleccionando la EIP y, desde el menú de “Acciones”, seleccionando la opción “Asociar la dirección IP elástica”:



En la siguiente ventana, seleccionamos como “Tipo de recurso” el valor “Instancia”, y elegimos nuestra instancia en ejecución. Por último, presionamos el botón “Asociar”.

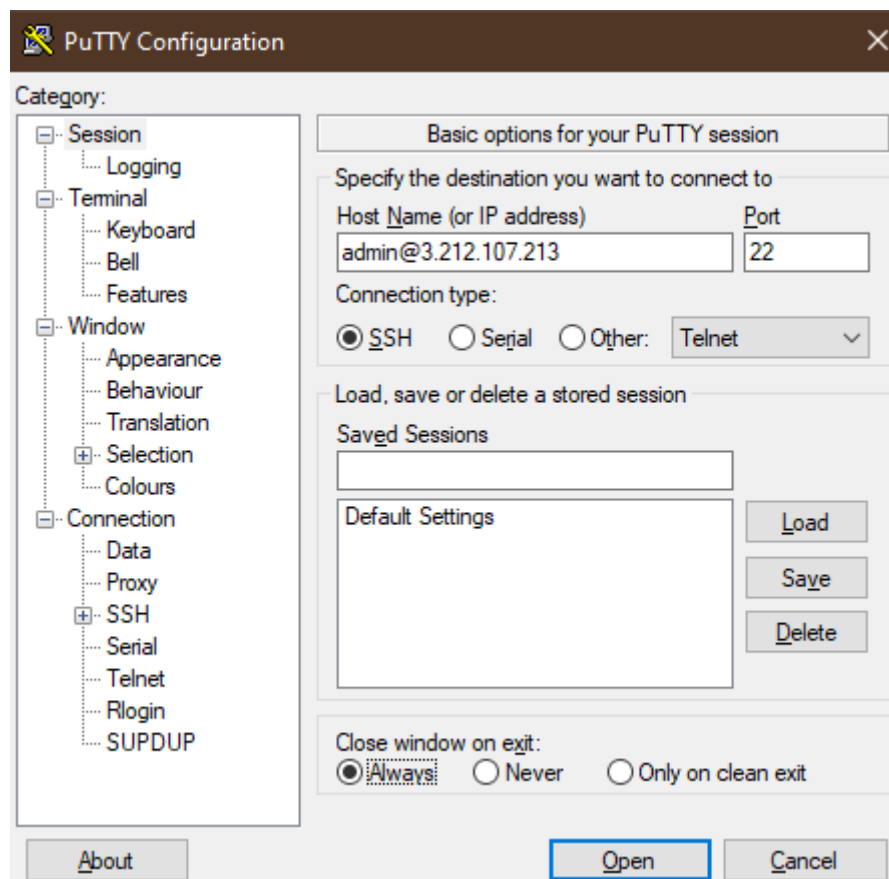


Podemos comprobar que nuestra instancia ya dispone de EIP:

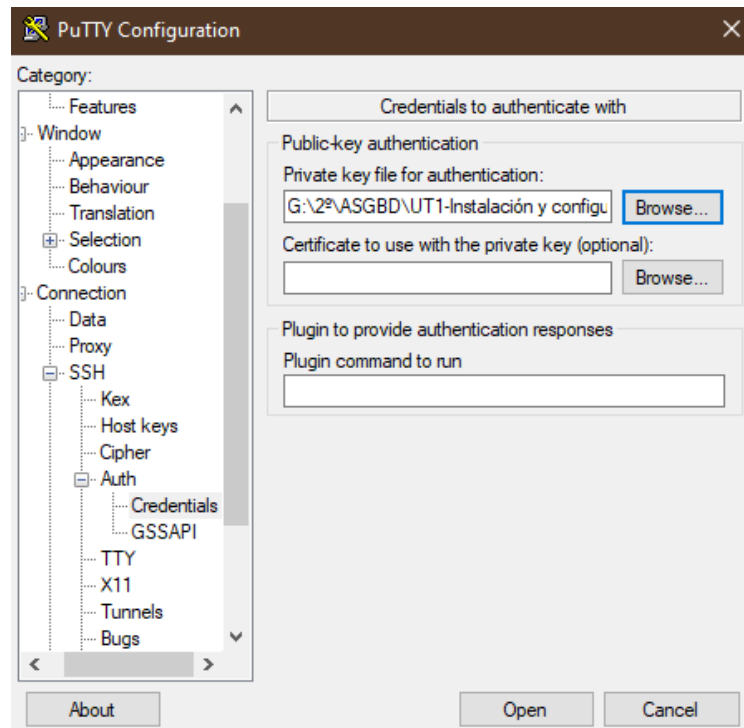


3. CONECTARTE POR SSH AL SERVIDOR E INSTALAR MARIADB

Vamos a conectarnos al servidor desde **Putty**. En "**Host Name (or IP address)**" escribimos el nombre de usuario, en este caso "**admin**", que es el usuario que crea AWS, con la EIP asociada:



Desde “SSH”, abrimos el menú de “Auth” y seleccionamos “Credentials”. Desde la nueva ventana, hacemos click en “Browse” en el cuadro de “Private key file for authentication” para seleccionar la clave que nos hemos descargado anteriormente. Después, presionamos el botón “Open”:



Ya estamos dentro de nuestro servidor:

```
admin@ip-172-31-13-44: ~
Using username "admin".
Authenticating with public key "imported-openssh-key"
Linux ip-172-31-13-44 6.1.0-10-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.37-1 (2023-07-03) x86_64

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

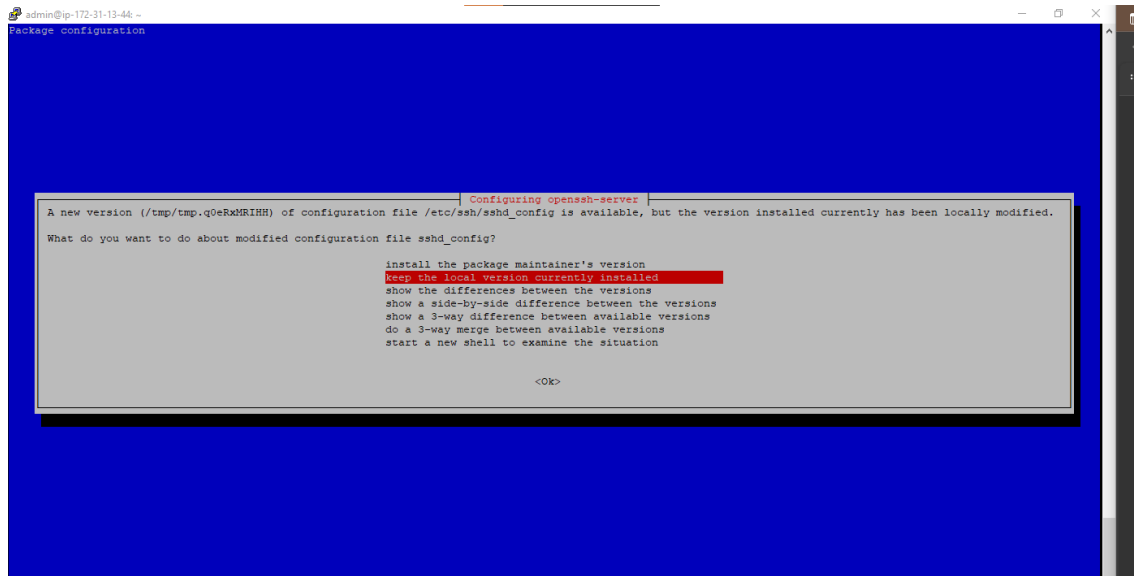
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@ip-172-31-13-44:~$
```

Antes de instalar nada, un *apt update* y un *apt upgrade* para actualizar el sistema:

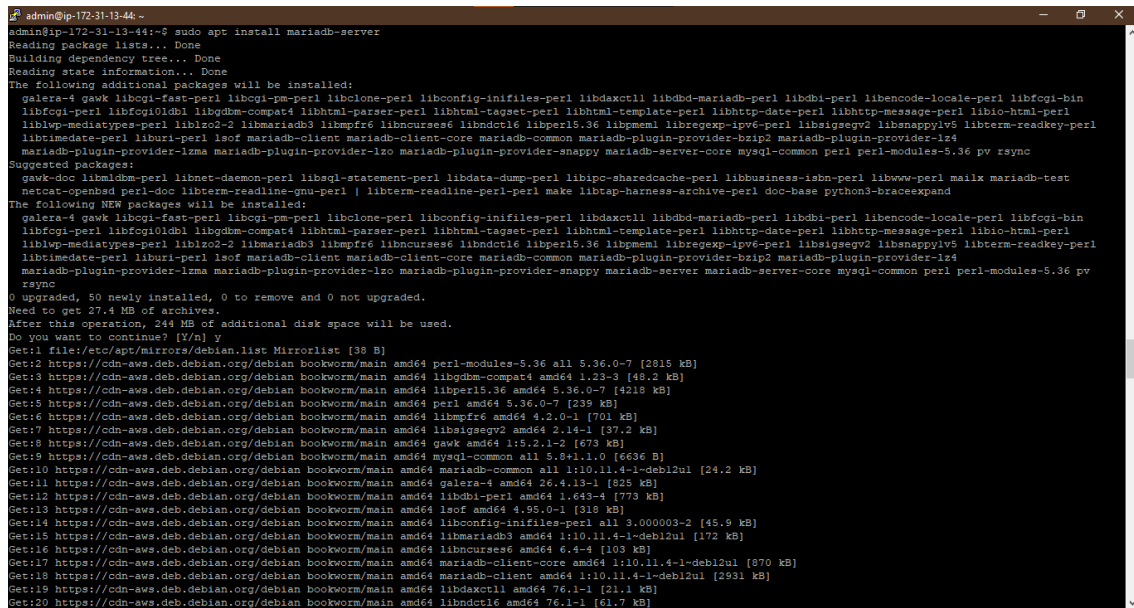
```
admin@ip-172-31-13-44:~$ sudo apt update
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [38 B]
Get:5 file:/etc/apt/mirrors/debian-security.list Mirrorlist [47 B]
Get:2 https://cdn-aws.deb.debian.org/debian bookworm InRelease [151 kB]
Get:3 https://cdn-aws.deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:4 https://cdn-aws.deb.debian.org/debian bookworm-backports InRelease [56.5 kB]
Get:6 https://cdn-aws.deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:7 https://cdn-aws.deb.debian.org/debian bookworm/main Sources [9488 kB]
Get:8 https://cdn-aws.deb.debian.org/debian bookworm/main amd64 Packages [8780 kB]
Get:9 https://cdn-aws.deb.debian.org/debian bookworm/main Translation-en [6110 kB]
Get:10 https://cdn-aws.deb.debian.org/debian bookworm-updates/main Sources [2324 B]
Get:11 https://cdn-aws.deb.debian.org/debian bookworm-updates/main amd64 Packages [6408 B]
Get:12 https://cdn-aws.deb.debian.org/debian bookworm-updates/main Translation-en [5008 B]
Get:13 https://cdn-aws.deb.debian.org/debian bookworm-backports/main Sources [116 kB]
Get:14 https://cdn-aws.deb.debian.org/debian bookworm-backports/main amd64 Packages [118 kB]
Get:15 https://cdn-aws.deb.debian.org/debian bookworm-backports/main Translation-en [97.6 kB]
Get:16 https://cdn-aws.deb.debian.org/debian-security bookworm-security/main Sources [51.5 kB]
Get:17 https://cdn-aws.deb.debian.org/debian-security bookworm-security/main amd64 Packages [86.2 kB]
Get:18 https://cdn-aws.deb.debian.org/debian-security bookworm-security/main Translation-en [48.8 kB]
Fetched 25.2 MB in 4s (6019 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
admin@ip-172-31-13-44:~$ sudo apt upgrade
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-6.1.0-19-cloud-amd64
/etc/kernel/postinst.d/sz-update-grub:
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.1.0-13-cloud-amd64
Found initrd image: /boot/initrd.img-6.1.0-13-cloud-amd64
Found linux image: /boot/vmlinuz-6.1.0-10-cloud-amd64
Found initrd image: /boot/initrd.img-6.1.0-10-cloud-amd64
done
Setting up libkrb5-3:amd64 (1.20.1-2+deb12u1) ...
Setting up qemu-utils (1:7.2+dfsg-7+deb12u2) ...
Setting up dbus-system-bus-common (1.14.10-1+deb12u1) ...
Setting up grub-efi-amd64-bin (2.06-13+deb12u1) ...
Setting up openssh (3.0.11-1+deb12u1) ...
Setting up libxml2:amd64 (2.9.14+dfsg-1.3+deb12u1) ...
Setting up dbus-bin (1.14.10-1+deb12u1) ...
Setting up grub-efi-amd64-signed (1:2.06+13+deb12u1) ...
Setting up grub2-common (2.06-13+deb12u1) ...
Setting up dbus-daemon (1.14.10-1+deb12u1) ...
Setting up grub-pc-bin (2.06-13+deb12u1) ...
Setting up dbus (1.14.10-1+deb12u1) ...
A reboot is required to replace the running dbus-daemon.
Please reboot the system when convenient.
dbus.service is a disabled or a static unit, not starting it.
Setting up libgssapi-krb5-2:amd64 (1.20.1-2+deb12u1) ...
Setting up linux-image-cloud-amd64 (6.1.55-1) ...
Setting up libpam-systemd:amd64 (252.17-1+deb12u1) ...
Setting up libcurl4:amd64 (7.88.1-10+deb12u4) ...
Setting up curl (7.88.1-10+deb12u4) ...
Setting up systemd-resolved (252.17-1+deb12u1) ...
Setting up bind9-libs:amd64 (1:9.18.19-1+deb12u1) ...
Setting up openssh-client (1:9.2p1-2+deb12u1) ...
Setting up libcurl3-gnutls:amd64 (7.88.1-10+deb12u4) ...
Setting up bind9-host (1:9.18.19-1+deb12u1) ...
Setting up libnss-resolve:amd64 (252.17-1+deb12u1) ...
Setting up openssh-sftp-server (1:9.2p1-2+deb12u1) ...
Setting up openssh-server (1:9.2p1-2+deb12u1) ...
rescue-ssh.target is a disabled or a static unit not running, not starting it.
ssh.socket is a disabled or a static unit not running, not starting it.
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for grub-cloud-amd64 (0.0.5) ...
Installing for i386-pc platform.
```

Con el **upgrade** nos aparecerá la siguiente ventana. Seleccionamos la opción que viene por defecto:



Una vez realizadas las actualizaciones, instalamos **MariaDB**:



Una vez instalamos, activamos el servicio:

```
admin@ip-172-31-13-44:~$ sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
admin@ip-172-31-13-44:~$ sudo systemctl start mariadb
admin@ip-172-31-13-44:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11.4 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Fri 2023-10-20 21:18:34 UTC; 2min 22s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 10820 (mariadb)
   Status: "Taking your SQL requests now..."
    Tasks: 10 (limit: 1144)
   Memory: 192.0M
      CPU: 439ms
   CGroup: /system.slice/mariadb.service
           └─10820 /usr/sbin/mariadb

Oct 20 21:18:34 ip-172-31-13-44 mariadb[10820]: 2023-10-20 21:18:34 0 [Note] Plugin 'FEEDBACK' is disabled.
Oct 20 21:18:34 ip-172-31-13-44 mariadb[10820]: 2023-10-20 21:18:34 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
Oct 20 21:18:34 ip-172-31-13-44 mariadb[10820]: 2023-10-20 21:18:34 0 [Note] Server socket created on IP: '127.0.0.1'.
Oct 20 21:18:34 ip-172-31-13-44 mariadb[10820]: 2023-10-20 21:18:34 0 [Note] InnoDB: Buffer pool(s) load completed at 231020 21:18:34
Oct 20 21:18:34 ip-172-31-13-44 mariadb[10820]: 2023-10-20 21:18:34 0 [Note] /usr/sbin/mariadb: ready for connections.
Oct 20 21:18:34 ip-172-31-13-44 mariadb[10820]: Version: '10.11.4-MariaDB-1-deb12u1' socket: '/run/mysqld/mysqld.sock' port: 3306 Debian 12
Oct 20 21:18:34 ip-172-31-13-44 systemd[1]: Started mariadb.service - MariaDB 10.11.4 database server.
Oct 20 21:18:34 ip-172-31-13-44 /etc/mysql/debian-start[10835]: Upgrading MySQL tables if necessary.
Oct 20 21:18:34 ip-172-31-13-44 /etc/mysql/debian-start[10846]: Checking for insecure root accounts.
Oct 20 21:18:34 ip-172-31-13-44 /etc/mysql/debian-start[10850]: Triggering mysam-recover for all MyISAM tables and aria-recover for all Aria tables
since 1-13/23 (END)
```

Ahora, lo configuramos:

```
admin@ip-172-31-13-44:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!
```

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
admin@ip-172-31-13-44:~$
```

4. CREAR UN USUARIO PERSONALIZADO CON TODOS LOS PERMISOS

Primero, accedemos a MariaDB:

```
admin@ip-172-31-13-44: ~
admin@ip-172-31-13-44:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.11.4-MariaDB-1~deb12ul Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Después, creamos el usuario con su respectiva contraseña:

```
MariaDB [(none)]> create user "andrea"@"localhost" identified by "Clave_00";
Query OK, 0 rows affected (0.002 sec)
```

Una vez creado, pasamos a darle todos los permisos:

```
MariaDB [(none)]> grant all privileges on *.* to "andrea"@"localhost" identified  
by "Clave_00" with grant option;  
Query OK, 0 rows affected (0.001 sec)
```

Ya podemos iniciar sesión con el nuevo usuario:

```
admin@ip-172-31-13-44: ~  
admin@ip-172-31-13-44:~$ mysql -u andrea -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 40  
Server version: 10.11.4-MariaDB-1~deb12u1 Debian 12  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> █
```