

Instalar MariaDB

1.- La instalación la hacemos en docker. Partimos que se ha instalado SSH, Apache2 y PHP. Aunque bastaría con tener instalado solo SSH.

```
usuario@usuario-VirtualBox:~$ docker commit c1_ssh_web imagenssh_mariadb  
sha256:bbd8340fa64f7570c4e331d2ca44a6c4f4b68fb4a76cb218e8a1b8b2ea40bab4  
usuario@usuario-VirtualBox:~$ docker stop c1_ssh_web  
c1_ssh_web  
usuario@usuario-VirtualBox:~$ docker rm c1_ssh_web  
c1_ssh_web  
usuario@usuario-VirtualBox:~$ docker run --name c1_ssh_db -it -d -p 2222:22 -p 3306:3306 imagenssh_mariadb  
668cdda09ac11b9bc661a59b88ce15d130e58886e2a011392066ce0c0543ceac  
usuario@usuario-VirtualBox:~$
```

Se ha creado una imagen 'imagenssh_mysql' que ya incluye ssh del ejercicio anterior y sobre la que instalaremos 'mariadb'.

El contenedor creado es 'c1_ssh_db'

```
~$ docker run --name c1_ssh_db -it -d -p 2222:22 -p 3306:3306 imagenssh_mariadb
```

Evidentemente Mariadb no esta ni instalada ni reiniciado el servicio. Por lo tanto, no se ofrece servicio alguno por el puerto 3306.

Desde el terminal ssh nos conectamos para proceder a la instalación de mariadb: previamente debemos entrar en el contenedor e iniciar el demonio SSH.

```
usuario@usuario-VirtualBox:~$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS  
668cdda09ac1   imagenssh_mariadb "bash"                  8 minutes ago Up 8 minutes       80/tcp, 0.0.0.0:3306->3306/tcp,  
:::3306->3306/tcp, 0.0.0.0:2222->22/tcp, :::2222->22/tcp  
e43763a71ab1   ubuntu        "bash"                  47 hours ago  Exited (0) 21 hours ago  
heuristic_engelbart  
usuario@usuario-VirtualBox:~$ docker exec -it c1_ssh_db /bin/bash  
root@668cdda09ac1:/# service ssh restart  
* Restarting OpenBSD Secure Shell server sshd  
root@668cdda09ac1:/#
```

Preferencias del escritorio remoto

Nombre: Conexión rápida

Grupo: [dropdown]

Protocolo: SSH - Secure Shell

Orden previa: command %h %u %t %U %p %g -option

Orden posterior: /path/to/command -opt1 arg %h %u %t -opt2 %U %p %g

Básico | Avanzado | Inicio automático | Túnel SSH

Servidor: 192.168.1.115:2222

Nombre de usuario: usuario

Contraseña de usuario: *****

Tipo de autenticación: Contraseña

Archivo de identidad: [checkbox] (Ninguno)

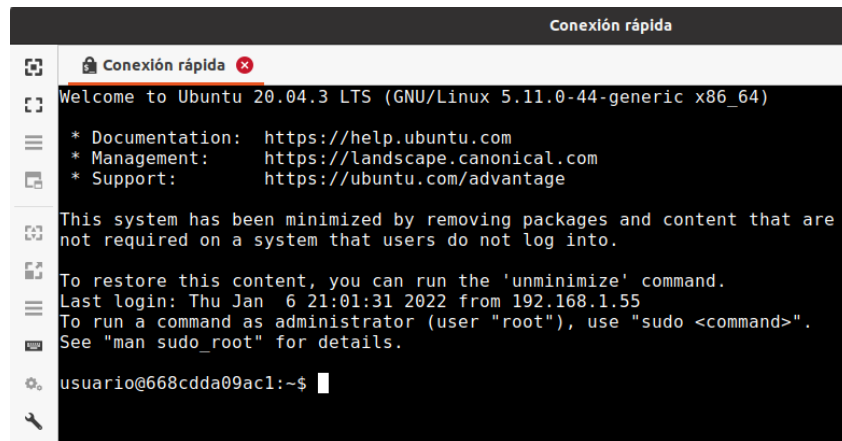
Contraseña para desbloquear la clave privada: [empty field]

Programa de inicio: [empty field]

Cancelar | Guardar como predeterminado | Guardar | Conectar | Guardar y conectar

Reiniciamos el servicio ssh:

```
root@804cae76681b:/# service ssh start
```



Procedemos a la instalación de MariaDB:

<https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-ubuntu-20-04-es>

Paso 1: Instalar MariaDB

En el momento de escribir este tutorial, los repositorios APT predeterminados de Ubuntu 20.04 incluyen la versión de MariaDB 10.3.

Para instalarla, actualice el índice de paquetes en su servidor con `apt`:

```
sudo apt update
```

A continuación, instale el paquete:

```
sudo apt install mariadb-server
```

Estos comandos instalarán MariaDB, pero no le solicitará establecer una contraseña ni realizar otros cambios de configuración. Debido a que la configuración predeterminada hace que la instalación de MariaDB no sea segura, usaremos una secuencia de comandos que el paquete de `mariadb-server` proporciona para restringir el acceso al servidor y eliminar cuentas no utilizadas.

Si fuese necesario instalar también:

```
/# apt install dialog
```

```
/# apt install apt-utils
```

Paso 2: Configurar MariaDB

En las nuevas instalaciones de MariaDB, el siguiente paso es ejecutar la secuencia de comandos de seguridad incluida. Esta secuencia de comandos cambia algunas de las opciones predeterminadas menos seguras para cosas como inicios de sesión **root** remotos y usuarios de ejemplo.

Ejecute la secuencia de comandos de seguridad:

```
sudo mysql_secure_installation
```

Con esto, verá una serie de solicitudes mediante las cuales podrá realizar cambios en las opciones de seguridad de su instalación de MariaDB. En la primera solicitud se pedirá que introduzca la contraseña **root** de la base de datos actual. Debido a que no configuramos una aún, **pulse ENTER** para indicar “none” (ninguna).

Output

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 you haven't set the root password yet, the password will be blank, so you should just press enter here.

Enter current password for root (enter for none):

En la siguiente solicitud se pregunta si desea configurar una contraseña **root** de base de datos. En Ubuntu, la cuenta **root** para MariaDB está estrechamente vinculada al mantenimiento del sistema automatizado. Por lo tanto, no deberíamos cambiar los métodos de autenticación configurados para esa cuenta. Hacer esto permitiría que una actualización de paquetes dañara el sistema de bases de datos eliminando el acceso a la cuenta administrativa. **Escriba N y pulse ENTER.**

Si tiene un ERROR para introducir el password del root:

To solve this issue you need to stop mysql process either through:

```
systemctl stop mysql      or      service mysql stop
```

after than you' ll need to expose the directory which has the socket configurations to the user who is running the process, this can happen through:

```
sudo usermod -d /var/lib/mysql/ mysql    <-----
```

then go back and start / restart mysql process again through:

```
systemctl start mysql      or      service start mysql
```

Output

```
. . .  
OK, successfully used password, moving on...
```

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

Set root password? [Y/n] N

Más tarde, se explicará la manera configurar una cuenta administrativa adicional para el acceso con contraseña si la autenticación del socket no es apropiada para su caso de uso.

Desde allí, puede **pulsar Y y luego ENTER** para aceptar los valores predeterminados **para todas las preguntas siguientes**. Con esto, se eliminarán algunos usuarios anónimos y la base de datos de

prueba, se deshabilitarán las credenciales de inicio de sesión remoto de **root** y se cargarán estas nuevas reglas para que MariaDB aplique de inmediato los cambios que realizó.

Con eso, ha terminado de realizar la configuración de seguridad inicial de MariaDB. El siguiente paso es opcional, aunque debería seguirlo si prefiere autenticar a su servidor de MariaDB con una contraseña.

Paso 3: (Opcional) Crear un usuario administrativo que emplea la autenticación por contraseña

En los sistemas Ubuntu con MariaDB 10.03, el **root** user de MariaDB se configura para autenticar usando el complemento `unix_socket` por defecto en vez de con una contraseña. Esto proporciona una mayor seguridad y utilidad en muchos casos, pero también puede generar complicaciones cuando necesita otorgar derechos administrativos a un programa externo (por ejemplo, `phpMyAdmin`).

En vez de modificar la cuenta **root**, los mantenedores de paquetes recomiendan crear una cuenta administrativa independiente para el acceso basado en contraseña.

Para hacerlo, crearemos una nueva cuenta llamada **admin** con las mismas capacidades que la cuenta **root**, pero configurada para la autenticación por contraseña. Abra la línea de comandos de MariaDB desde su terminal:

```
$ sudo mariadb
```

A continuación, cree un nuevo usuario con privilegios **root** y acceso basado en contraseña. Asegúrese de cambiar el nombre de usuario y la contraseña para que se adapten a sus preferencias:

```
GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;
```

Vacíe los privilegios para garantizar que se guarden y estén disponibles en la sesión actual:

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Después de esto, cierre el shell de MariaDB:

```
MariaDB [(none)]> exit
```

Por último, probaremos la instalación de MariaDB.

Paso 4: Probar MariaDB

Cuando se instale desde los repositorios predeterminados, MariaDB se ejecutará automáticamente. Para probar esto, compruebe su estado.

```
$ sudo service mysql start | stop | restart | status    ///   service mariadb start.....
```

```
sudo systemctl status mysql
```

Como comprobación adicional, puede intentar establecer conexión con la base de datos usando la herramienta `mysqladmin`, que es un cliente que le permite ejecutar comandos administrativos. Por ejemplo, este comando indica conectar con MariaDB como **root** usando el socket Unix y devolver la versión:

```
sudo mysqladmin version
```

Si configuró un usuario administrativo independiente con la autenticación de contraseña, puede realizar la misma operación escribiendo lo siguiente:

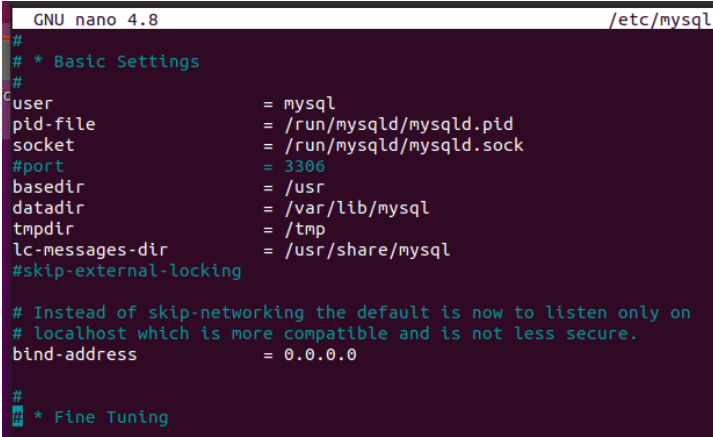
```
mysqladmin -u admin -p version
```

Esto significa que MariaDB está activo y que su usuario puede autenticarse correctamente ya que solo está mostrando la versión.

//// opcional si hay error Crear archivo en la siguiente carpeta: `touch mysql.sock`

//// opcional si hay error - Cambiar los permisos: `chown mysql:mysql /var/run/mysqld/mysqld.sock`

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```



```
GNU nano 4.8 /etc/mysql/
#
# * Basic Settings
#
user                = mysql
pid-file            = /run/mysqld/mysqld.pid
socket              = /run/mysqld/mysqld.sock
#port               = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc-messages-dir     = /usr/share/mysql
#skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 0.0.0.0

#
# * Fine Tuning
```

Aseguramos una imagen de docker con todas la instalacion realizada

```
usuario@usuario-VirtualBox:~$ docker commit c1_ssh_db imagenssh_mariadb
```

**** reconstruir una imagen por seguridad**

```
..$ docker commit e43763a71ab1 imagenssh
```

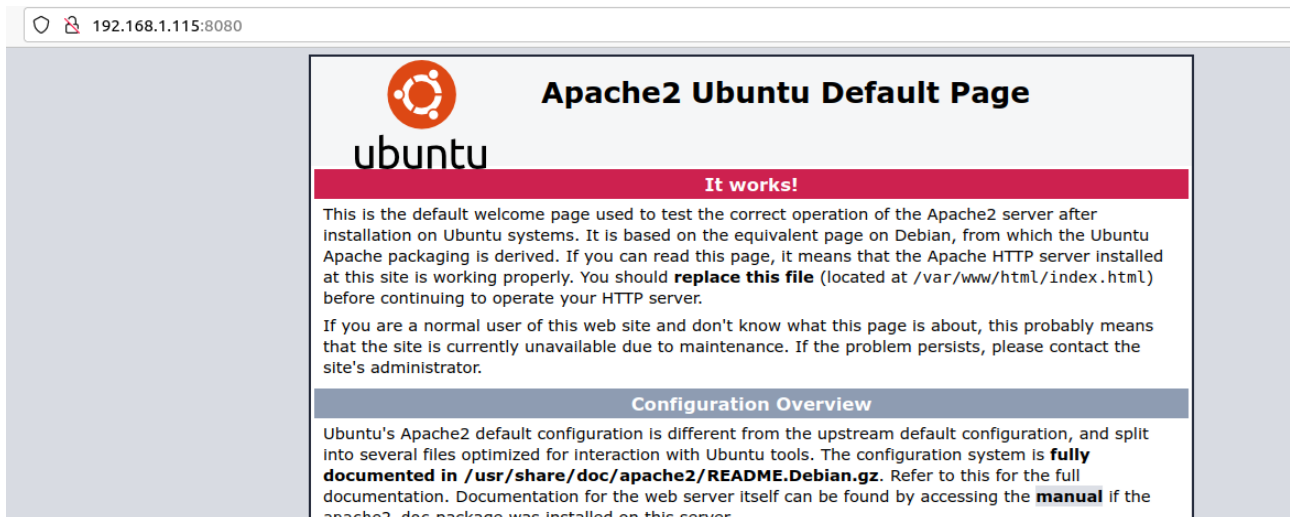
Ahora si podremos lanzar un contenedor desde la imagen 'imagenssh' y habilitar el puerto 22

```
..$ docker run --name c1_ssh_web -it -d -p 2222:22 -p 8080:80 imagenssh_web
```

Para abrir un terminal del contenedor para activar ssh-server

```
..$ docker exec -it c1_ssh_web /bin/bash
```

Comprobamos la ejecución del servidor web escribiendo la **IP:8080** en el navegador:



Ahora todos los archivos de nuestra pagina web que colocamos en /var/www/html (caso de Ubuntu) son los que visualiza el servidor web. Como llevarlos hasta allí se realizaría con un servidor FTP.

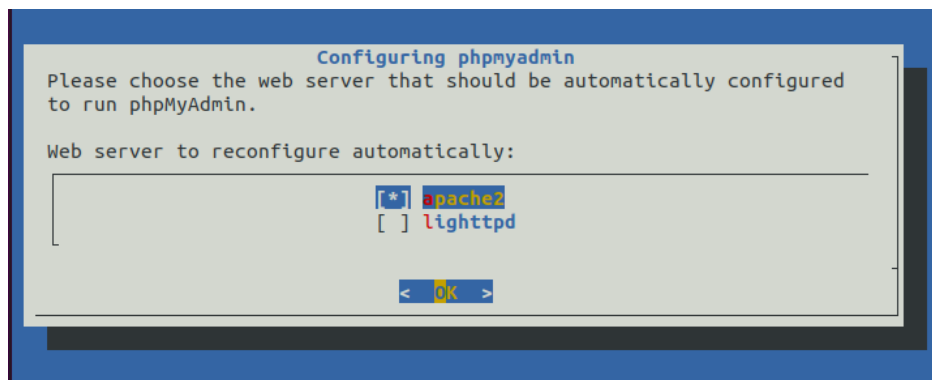
Pero en nuestro caso lo que nos interesa en tener instalado un servidor de Base de Datos (mariadb) y como conectarla a nuestra web..

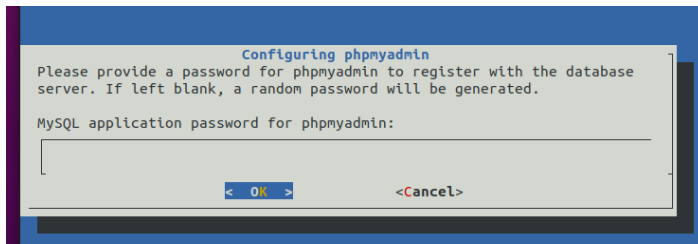
Se puede hacer copia de seguridad con un Bind Mount sobre la carpeta del ordenador:

```
..$ docker run -d -it -p 2222:22 -p 3306:3306 -v /home/usuario/miweb1/miweb1_db:/var/lib/mysql --name c2_ssh_db imagenssh_mariadb
```

Si instalamos phpmyadmin:

```
sudo apt install phpmyadin
```

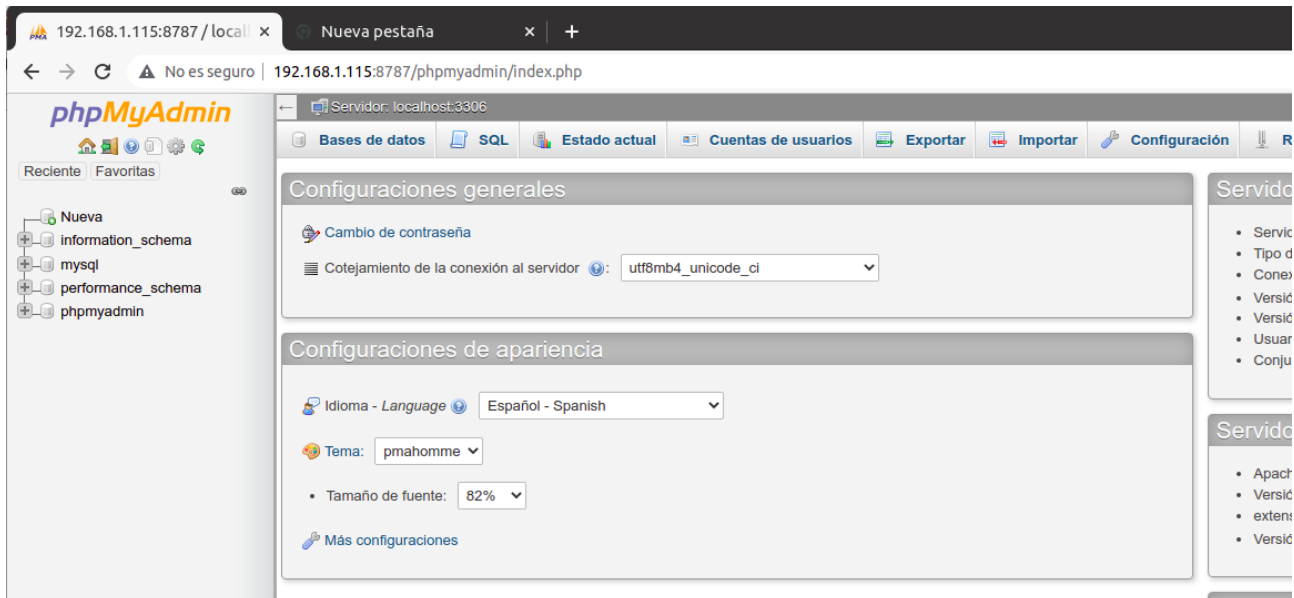




introduce tu password
repite de nuevo tu password

Reinicia apache para que tome la instalacion de phpmyadmin:
service apache2 restart

desde la maquina virtual:
http://localhost:8787/phpmyadmin
usuario: admin
password: tu-password



```
docker run -d -it -p 2222:22 -p 3336:3306 -p 8787:80 -v --name c3_ssh_db_web  
/home/usuario/miweb1/miweb1_db:/var/lib/mysql imagenssh_mariadb
```

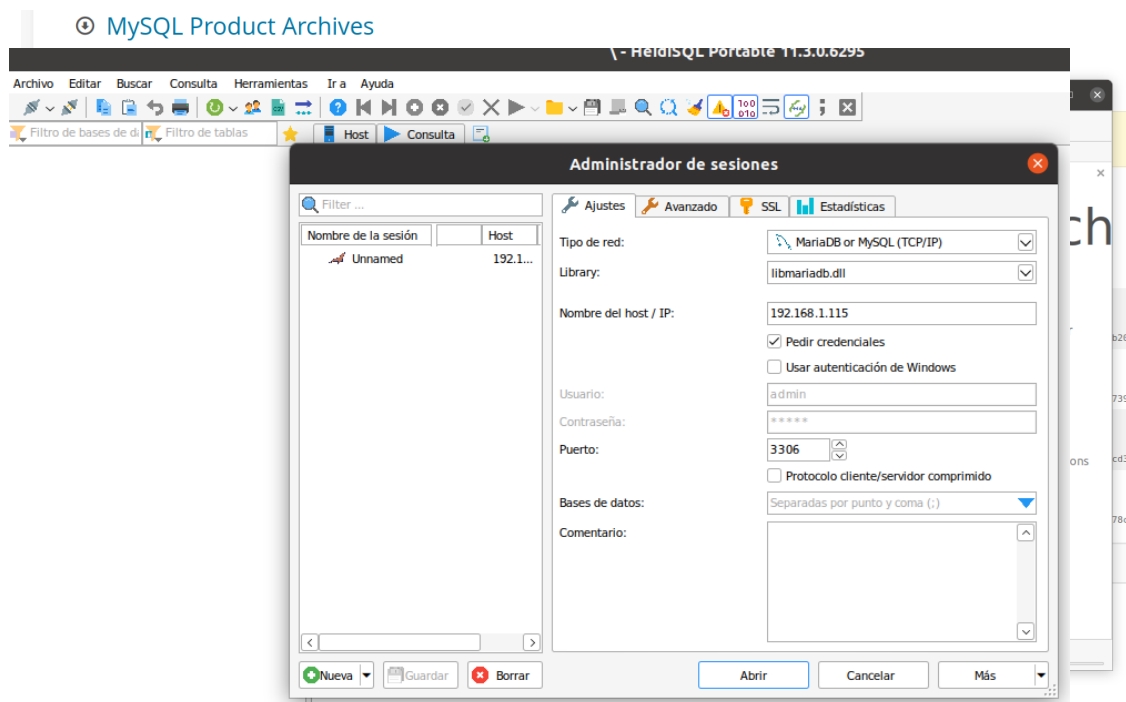
Conectar con workbench /Heidi:
1.- Descargar/Instalar
2.- Conectar

Creamos un usuario de acceso remoto:

```
GRANT ALL ON *.* TO 'admin'@'%' IDENTIFIED BY 'password' WITH GRANT OPTION;
```

Podemos crear un usuario de acceso remoto desde un equipo determinado:

```
GRANT ALL ON *.* TO 'admin'@'192.168.1.115' IDENTIFIED BY 'password' WITH GRANT OPTION;
```



Resumen de comandos de instalación

```
apt update  
apt upgrade
```

```
apt install net-tools  
apt install iproute2
```

Crear usuario

```
apt install openssh-server  
apt install sudo  
apt install dialog  
apt install apache2  
apt install apt-utils  
apt install php  
service ssh start  
service apache2 start
```

```
sudo apt install mariadb-server
```



```
sudo usermod -d /var/lib/mysql/ mysql
sudo mysql_secure_installation (enter, N, Y, Y, Y...)
```

```
sudo mariadb
```

```
GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;
```

```
GRANT ALL ON *.* TO 'admin'@'%' IDENTIFIED BY 'admin' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

```
exit
```

```
service mysql restart
```

```
root@4ebe2a693f43:/# netstat -ltn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22               0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80               0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                     LISTEN
root@4ebe2a693f43:/#
```

```
apt install phpmyadmin
```

```
service apache2 restart
```