

# IMPLANTACIÓN DE APLICACIONES WEB

*Jose Materano*

Actividad 2 Manual de Despliegue de Entorno LAMP con acceso remoto a las bases de datos desde cualquier cliente

## TABLA DE CONTENIDOS

<b>Act 2. Manual de Despliegue de WordPress y Drupal en Docker .....</b>	<b>3</b>
1. 2. Requisitos Previos.....	3
2. 3. Actualizar el Sistema.....	4
3. 4. Instalar Apache.....	4
4. 5. Configurar el Grupo de Seguridad de AWS.....	5
5. 6. Configurar Apache para Escuchar en el Puerto 8082.....	5
6. 7. Instalar MySQL .....	6
7. 8. Instalar PHP.....	7
8. 9. Crear una Página de Prueba .....	7
9. 10. Crear una Página PHP para Verificar la Conexión a MySQL .....	8
10. 11. Verificar la Conexión a MySQL en el Navegador .....	9
11. 12. Verificar la Conexión a MySQL desde un cliente externo cualquiera .....	10

# Act 2. Manual de Despliegue de WordPress y Drupal en Docker

## 2. Requisitos Previos

Este documento describe el proceso paso a paso para desplegar un entorno LAMP (Linux, Apache, MySQL, PHP) completo utilizando una instancia de AWS. Al final del proceso, se comprobará la configuración mediante una página de prueba accesible en el puerto 8082.

- Sistema Operativo: Ubuntu Server

### Lanzar la instancia

Para comenzar, lance una instancia de Amazon EC2, que es un servidor virtual en la nube.

[Lanzar la instancia](#)[Migrar un servidor](#)

Nota: Sus instancias se lanzarán en la región Europa (París)

### Inicio rápido

Amazon Linux 	Ubuntu 	Windows 	Red Hat 	SUSE Linux 	Debian 	 <a href="#">Buscar más AMI</a> Inclusión de AMI de AWS, Marketplace y la comunidad
------------------	------------	-------------	-------------	----------------	------------	---

### Imágenes de máquina de Amazon (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type  
ami-07db896e164bc4476 (64 bits (x86)) / ami-0e1e669f1f5062039 (64 bits (Arm))  
Virtualización: hvm Activado para ENA: true Tipo de dispositivo raíz: ebs

Apto para la capa gratuita



- Conexión por ssh a nuestra maquina

Ejemplo:

```
ssh -i "LAMP_ACT.pem" ubuntu@ec2-13-38-24-116.eu-west-3.compute.amazonaws.com
```

Para evitar errores debemos ejecutar un cmd dentro de donde esta contenida nuestra clave.pem

### 3. Actualizar el Sistema

- **sudo apt update y sudo apt upgrade**

```
ubuntu@ip-172-31-0-80:~$ sudo apt update
Hit:1 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1956 kB]
Get:8 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [9772 B]
```

```
Unpacking linux-image-6.8.0-1019-aws (6.8.0-1019.21~22.04.1) ...
Preparing to unpack ../23-linux-aws_6.8.0-1019.21~22.04.1_amd64.deb ...
Unpacking linux-aws (6.8.0-1019.21~22.04.1) over (6.8.0-1015.16~22.04.1) ...
Preparing to unpack ../24-linux-image-aws_6.8.0-1019.21~22.04.1_amd64.deb ...
Unpacking linux-image-aws (6.8.0-1019.21~22.04.1) over (6.8.0-1015.16~22.04.1) ...
Selecting previously unselected package linux-aws-6.8-headers-6.8.0-1019.
Preparing to unpack ../25-linux-aws-6.8-headers-6.8.0-1019_6.8.0-1019.21~22.04.1_all.deb ...
Unpacking linux-aws-6.8-headers-6.8.0-1019 (6.8.0-1019.21~22.04.1) ...
```

```
Progress: [ 33%] [#####.....]
```

### 4. Instalar Apache

- Primero instalamos el servicio de Apache  
**sudo apt install apache2 -y**

```
ubuntu@ip-172-31-0-80:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
```

- Habilita Apache para que inicie al arrancar el sistema y nos aseguramos que esta corriendo:

**sudo systemctl enable apache2**

**sudo systemctl start apache2**

```
ubuntu@ip-172-31-0-80:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

## 5. Configurar el Grupo de Seguridad de AWS

En la consola de AWS, agrega una regla al grupo de seguridad asociado a la instancia para permitir el tráfico en el puerto 8082 desde cualquier lugar (0.0.0.0/0).

- **Tipo:** Custom TCP
- **Puerto:** 8082
- **Origen:** 0.0.0.0/0



## 6. Configurar Apache para Escuchar en el Puerto 8082

- Editamos el archivo de configuración de Apache para escuchar en el puerto 8082:  
`sudo nano /etc/apache2/ports.conf`

```
Listen 8082_

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

- Guardamos los cambios y reinicia Apache:  
`sudo systemctl restart apache2`  
`sudo systemctl status apache2`

```

ubuntu@ip-172-31-0-80:~$ sudo nano /etc/apache2/ports.conf
ubuntu@ip-172-31-0-80:~$ sudo systemctl restart apache2
ubuntu@ip-172-31-0-80:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-12-03 03:21:31 UTC; 8s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 13216 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 13222 (apache2)
    Tasks: 55 (limit: 1130)
   Memory: 5.2M
      CPU: 23ms
   CGroup: /system.slice/apache2.service
           └─13222 /usr/sbin/apache2 -k start
             13223 /usr/sbin/apache2 -k start
             13224 /usr/sbin/apache2 -k start

Dec 03 03:21:31 ip-172-31-0-80 systemd[1]: Starting The Apache HTTP Server...
Dec 03 03:21:31 ip-172-31-0-80 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-0-80:~$

```

## 7. Instalar MySQL

- Para instalar MySQL, ejecuta el siguiente comando:

**sudo apt install mysql-server -y**

```

ubuntu@ip-172-31-0-80:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbcbi-fast-perl libbcbi-pm-perl libclone-perl libencode-locale-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab

```

- Protegemos la instalación de MySQL ejecutando el script de configuración:

**sudo mysql\_secure\_installation**

Aquí le damos a todo que si a menos que nos sea de utilidad dejar alguna característica activada como el remote login

```

Remove anonymous users? (Press y|Y for Yes, any other key for No) : 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? (Press y|Y for Yes, any other key for No) : n
... skipping.
By default, MySQL 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? (Press y|Y for Yes, any other key for No) : 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? (Press y|Y for Yes, any other key for No) : y
Success.

All done!

```

## 8. Instalar PHP

- Instala PHP y los módulos necesarios para que funcione con Apache y MySQL:

```
sudo apt install php libapache2-mod-php php-mysql -y
```

```
ubuntu@ip-172-31-0-80:~$ sudo apt install php libapache2-mod-php php-mysql -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common php8.1-mysql p
Suggested packages:
```

- Verifica la instalación de PHP ejecutando:

```
php -v
```

```
ubuntu@ip-172-31-0-80:~$ php -v
PHP 8.1.2-1ubuntu2.19 (cli) (built: Sep 30 2024 16:25:25) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.19, Copyright (c), by Zend Technologies
```

## 9. Crear una Página de Prueba

Creemos un archivo PHP para probar la configuración de Apache y PHP:

```
echo '<?php phpinfo(); ?>' | sudo tee /var/www/html/info.php
```

Además, nos aseguramos de que el archivo se ha creado correctamente y de que Apache tiene permisos para servir el archivo.

```
ubuntu@ip-172-31-0-80:~$ sudo chmod -R 755 /var/www/html
ubuntu@ip-172-31-0-80:~$ echo '<?php phpinfo(); ?>' | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
ubuntu@ip-172-31-0-80:~$ ls /var/www/html/info.php
/var/www/html/info.php
ubuntu@ip-172-31-0-80:~$
```

## 10. Crear una Página PHP para Verificar la Conexión a MySQL

- Antes de comenzar nos aseguramos renovar nuestras credenciales y las renovamos

**Sudo mysql -u root**

```
ubuntu@ip-172-31-0-80:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '4.Vientos';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> EXIT
Bye
```

- Luego creamos un archivo llamado **mysql\_test.php** en el directorio **/var/www/html** para verificar que puedes conectarte al servidor MySQL:

**sudo nano /var/www/html/mysql\_test.php** y le pegamos este código sencillo

```
<?php

$servername = "localhost";

$username = "root";

$password = "4.Vientos";

// Crear conexión

$conn = new mysqli($servername, $username, $password);

// Verificar conexión

if ($conn->connect_error) {

    die("Conexión fallida: " . $conn->connect_error);

}

echo "Conexión exitosa a MySQL";

?>
```



```
GNU nano 6.2
<?php
$servername = "localhost";
$username = "root";
$password = "4.Vientos";
// Crear conexión
$conn = new mysqli($servername, $username, $password);
// Verificar conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
echo "Conexión exitosa a MySQL";
?>
```

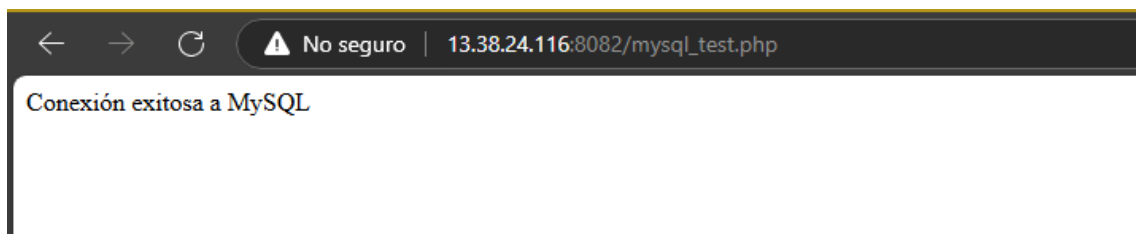
- Guardamos los cambios y cerramos el editor

## 11. Verificar la Conexión a MySQL en el Navegador

- Abrimos un navegador y accede a la dirección IP del host seguido del puerto 8082 para ver la página de prueba de MySQL:

[http://<IP\\_PUBLICA>:8082/mysql\\_test.php](http://<IP_PUBLICA>:8082/mysql_test.php)

Deberíamos ver el mensaje "Conexión exitosa a MySQL" si todo está correctamente configurado.



- Ahora verificamos la pagina que colocamos en nuestro apache

[http://<IP\\_PUBLICA>:8082/info.php](http://<IP_PUBLICA>:8082/info.php)

A screenshot of a web browser window displaying the PHP version and system information page. The address bar shows a warning icon, the text 'No seguro', and the URL '13.38.24.116:8082/info.php'. The page content includes the PHP logo and a table with system information.

PHP Version 8.1.2-1ubuntu2.19	
System	Linux ip-172-31-0-80 6.8.0-1015-aws #16~22.04.1-Ubuntu SMP Mon Aug 19 19:38:17 UTC 2024 x86_64
Build Date	Sep 30 2024 16:25:25
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d

- Ahora Verificamos los servicios para ver que todo esté en orden

**sudo systemctl status apache2**

**sudo systemctl status mysql**

```

ubuntu@ip-172-31-0-80:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-12-03 03:23:12 UTC; 26min ago
     Main PID: 13976 (mysqld)
    Status: "Server is operational"
       Tasks: 38 (limit: 1130)
      Memory: 351.6M
         CPU: 7.005s
        CGroup: /system.slice/mysql.service
                └─13976 /usr/sbin/mysqld

Dec 03 03:23:11 ip-172-31-0-80 systemd[1]: Starting MySQL Community Server...
Dec 03 03:23:12 ip-172-31-0-80 systemd[1]: Started MySQL Community Server.
ubuntu@ip-172-31-0-80:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-12-03 03:31:02 UTC; 19min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 21338 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
     Main PID: 21342 (apache2)
        Tasks: 6 (limit: 1130)
       Memory: 10.8M
          CPU: 78ms
        CGroup: /system.slice/apache2.service
                └─21342 /usr/sbin/apache2 -k start
                  └─21344 /usr/sbin/apache2 -k start
                    └─21345 /usr/sbin/apache2 -k start
                      └─21346 /usr/sbin/apache2 -k start
                        └─21347 /usr/sbin/apache2 -k start
                          └─21348 /usr/sbin/apache2 -k start

Dec 03 03:31:02 ip-172-31-0-80 systemd[1]: Starting The Apache HTTP Server...
Dec 03 03:31:02 ip-172-31-0-80 systemd[1]: Started The Apache HTTP Server.

```

## 12. Verificar la Conexión a MySQL desde un cliente externo cualquiera

Para comprobar que MySQL está funcionando correctamente, vamos a hacerlo desde un cliente remoto Heidi SQL:

- Primero, accede a MySQL:

**sudo mysql -u root -p**

```

ubuntu@ip-172-31-0-80:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

```

- Luego creamos un nuevo usuario que tenga acceso remoto

**CREATE USER 'mate'@'%' IDENTIFIED BY '4.Vientos';**

**GRANT ALL PRIVILEGES ON \*.\* TO 'mate'@'%' WITH GRANT OPTION;**

**FLUSH PRIVILEGES;**

```

owners.

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

mysql> CREATE USER 'mate'@'%' IDENTIFIED BY '4.Vientos';
;
FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'mate'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>

```

- Por defecto, MySQL solo escucha conexiones en localhost. Necesitamos cambiar esta configuración para permitir conexiones remotas.

**sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf**

```

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
#

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# 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
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack        = 256K

```

- Tenemos que buscar la línea que contiene bind-address. Por defecto, estará configurada para **127.0.0.1** y cambiarla para que escuche en todas las interfaces de red **0.0.0.0**:

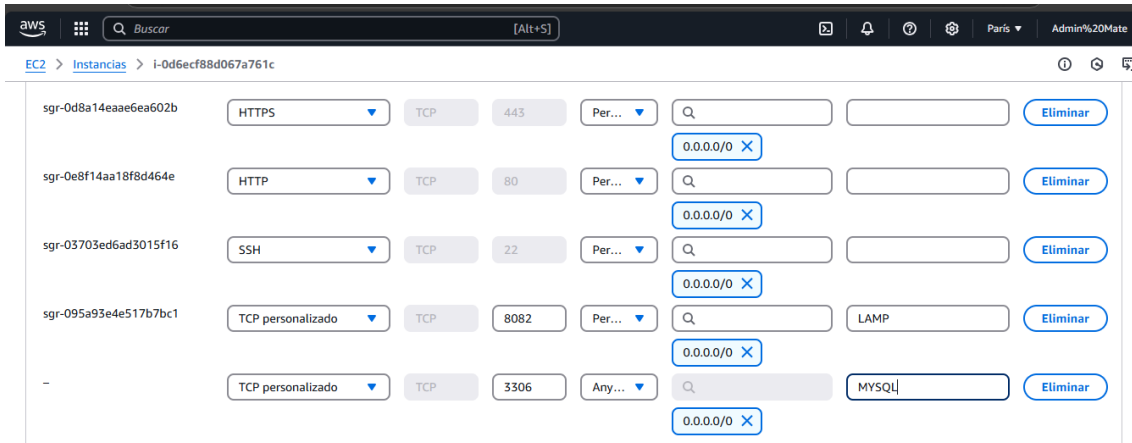
**bind-address = 0.0.0.0**

```

# localhost which is more compatible and is not less secure.
bind-address         = 0.0.0.0

```

- Luego reiniciamos el servicio SQL  
sudo systemctl restart mysql
- Añadimos una regla para permitir el tráfico SQL en nuestra instancia de AWS



- Abrimos HeidiSQL y creamos una nueva sesión:

Haz clic en "New" para crear una nueva conexión.

- Configurar la Conexión: Aquí un ejemplo de cómo llenar los campos principales para conectar con tu base de datos MySQL:
- Network type: Elige MySQL (TCP/IP).
- Hostname / IP: Coloca la IP pública de tu instancia EC2. Esta IP la puedes encontrar en la consola de AWS.
- User: Especifica el nombre de usuario que creaste para acceso remoto en mi caso mate
- Password: Ingresa la contraseña que estableciste para ese usuario en caso 4.Vientos.
- Port: Asegúrate de que el puerto sea 3306.

