**Tutorial del seteo de LAMP en Ubuntu 20.04**

**1.- Tener todo actualizado**

Primero nos fijaremos que tenemos todo completamente actualizado

sudo apt update

sudo apt upgrade

**2.- Instalar Apache**

Una vez este completamente actualizado pasamos a la siguiente fase que seria instalar Apache 2

sudo apt install -y apache2 apache2-utils

Para saber que fue instalado correctamente usaremos el siguiente comando

systemctl status apache2

Esto nos deberia mostrar algo aproximadamente parecido a lo siguiente:

● apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)

Active: active (running) since Sat 2020-04-11 11:31:31 CST; 2s ago

Docs: https://httpd.apache.org/docs/2.4/

Process: 53003 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)

Main PID: 53011 (apache2)

Tasks: 55 (limit: 19072)

Memory: 6.4M

CGroup: /system.slice/apache2.service

├─53011 /usr/sbin/apache2 -k start

├─53012 /usr/sbin/apache2 -k start

└─53013 /usr/sbin/apache2 -k start

Sino no corre, tendriamos que introducir el siguiente comando:

sudo systemctl start apache2

Apache es basicamente nuestro servidor donde estara tanto nuestras aplicaciones como tambien nuestro interprete php, perl y otros, ademas de nuestra base de datos, por esa razon necesitamos que este habilitado desde el inicio de sesion.

sudo systemctl enable apache2

Para poder visualizar la version de apache es nesesario el siguiente comando:

apache2 -v

Para poder acceder al servidor apache nos dirigimos a nuestro buscador (google o firefox) y colocamos cualquiera de estas 2 direcciones:

http://localhost

http://127.0.0.1

Estas direcciones son consideradas como loop, o home, es decir que nos sirven unicamente a nosotros para hacer pruebas locales.

Por defecto el servidor apache usa /var/www/html entonces tenemos que ownearlo

sudo chown www-data:www-data /var/www/html/ -R

**3.- Instalar MariaDB**

Utilizaremos el siguiente comando:

sudo apt install mariadb-server mariadb-client

de ahi verificamos su estado de instalacion:

systemctl status mariadb

Deberia salir que se encuentra activo:

● mariadb.service - MariaDB 10.3.22 database server

Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2020-04-10 14:19:16 UTC; 18s ago

Docs: man:mysqld(8)

https://mariadb.com/kb/en/library/systemd/

Main PID: 9161 (mysqld)

Status: "Taking your SQL requests now..."

Tasks: 31 (limit: 9451)

Memory: 64.7M

CGroup: /system.slice/mariadb.service

└─9161 /usr/sbin/mysqld

Si no se encuentra habilitada usamos el comando:

sudo systemctl start mariadb

Para poder hacer que inicie conjuntamente con el inicio de sesion colocamos:

sudo systemctl enable mariadb

**A continuacion toca una parte *sumamente importante***

Toca la instalacion segura como tal de mariadb

sudo mysql\_secure\_installation

para verificar que lo hicieron bien hacer lo siguiente:

sudo mariadb -u root

para salir del interprete de mariadb usan:

exit;

**3.- Instalacion de PHP 7**

**//si se necesita en 18.04**

**https://linuxize.com/post/how-to-install-php-on-ubuntu-18-04/**

Para instalar el interprete de php necesitamos usar el siguiente comando:

sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline

Necesitamos ahora habilitar php 7 en nuestro servidor

sudo a2enmod php7.4

sudo systemctl restart apache2

**4.- Instalar lo necesario para correr laravel**

sudo apt install php7.4-common php7.4-cli php7.4-gd php7.4-mysql php7.4-curl php7.4-intl php7.4-mbstring php7.4-bcmath php7.4-imap php7.4-xml php7.4-zip php7.4-pgsql

despues nos toca tener curl y otros detalles mas

sudo apt install -y git curl wget zip unzip

nos bajaremos composer que es el manejador de dependencias de laravel

curl -sS https://getcomposer.org/installer | php

de ahi lo moveremos a su lugar correcto

sudo mv composer.phar /usr/local/bin/composer

lo volvemos ejecutable

sudo chmod +x /usr/local/bin/composer

y probamos:

composer --version

cuando tengamos un proyecto de laravel descargado desde github lo unico que tenemos que hacer es:

composer install

y esto hara su magia sola

**LARAVEL**

**PHPMYADMIN**

sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl

**apretar espacio para que salga un \***

sudo mariadb -u root -p

use mysql;

update user set authentication\_string=PASSWORD("contraseña") where user = 'root';

update user set plugin='mysql\_native\_password' where user='root';

flush privileges;

exit;

**por si llega a fallar el inicio**

https://www.enmimaquinafunciona.com/pregunta/74161/como-acceder-a-phpmyadmin-en-la-lampara

**LARAVEL**

cd /var/www/html

sudo chown -R $USER:root /var/www

sudo rm index.html

git clone https://github.com/RobertoDominguez/e-commerce.git

cd carpeta

composer install

sudo cp .env.example .env

sudo mv server.php index.php

sudo php artisan key:generate

cd ..

sudo chmod -R 755 html

cd html

sudo chmod -R 777 storage

sudo a2enmod rewrite

sudo nano /etc/apache2/apache2.conf

(copiar)

<Directory /var/www/html/>

Options Indexes FollowSymLinks

AllowOverride all

Require all granted

</Directory>

**(En var/www/html)**

sudo nano .htaccess

<IfModule mod\_rewrite.c>

<IfModule mod\_negotiation.c>

Options -MultiViews

</IfModule>

RewriteEngine On

# Redirect Trailing Slashes If Not A Folder...

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^(.)/$ /$1 [L,R=301]

# Handle Front Controller...

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^ index.php [L]

# Handle Authorization Header

RewriteCond %{HTTP:Authorization} .

RewriteRule . - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]

</IfModule>

sudo nano .env

ASSET\_URL=/public

sudo php artisan storage:link

sudo php artisan config:cache

sudo systemctl restart apache2

localhost

Route::get('/foo', function () {

Artisan::call('storage:link');

});

Route::get('/clear-cache', function() {

$run = Artisan::call('config:clear');

$run = Artisan::call('cache:clear');

$run = Artisan::call('config:cache');

return 'FINISHED';

});

Contraseña phpmyadmin 12345

**Para Php 7.4**

sudo apt -y install software-properties-common

sudo add-apt-repository ppa:ondrej/php

sudo apt-get install php-gd php-xml php7.4-mbstring

sudo apt install php7.4-xml

sudo apt-get install php7.4-gd

# pgsql

sudo apt install php7.4-pgsql

sudo apt install postgresql postgresql-contrib

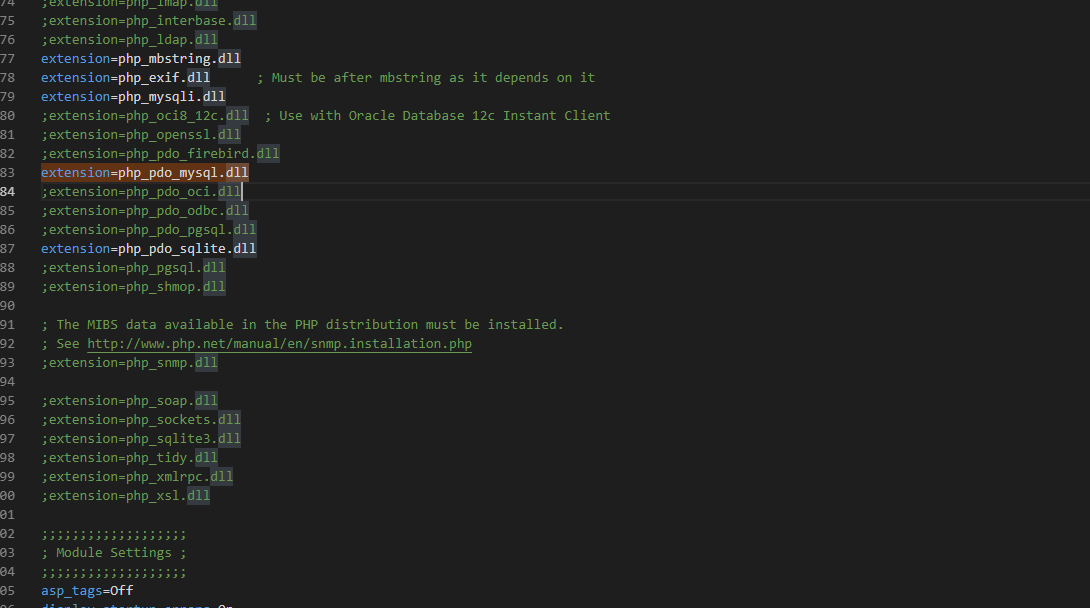
psql

alter user postgres with password 'passwd';

create database “nombre”;

exit;

nano /etc/php/7.4/apache2/php.ini



sudo apt-get **install php**-**imagick**

## Paso 1: Instalar PostgreSQL

Los repositorios predeterminados de Ubuntu contienen paquetes de Postgres, para que pueda instalarlos utilizando el sistema de empaquetado apt.

Si no lo hizo recientemente, actualice el índice del paquete local de su servidor:

sudo apt update

Copy

Luego, instale el paquete de Postgres junto con un paquete -contrib, que agrega algunas utilidades y funcionalidades adicionales:

sudo apt install postgresql postgresql-contrib

Copy

Ahora que el software está instalado, podemos analizar su funcionamiento y la forma en que puede diferenciarse de otros sistemas de administración de bases de datos relacionales que pueda haber utilizado.

## Paso 2: Utilizar roles y bases de datos de PostgreSQL

Por defecto, Postgres utiliza un concepto llamado “roles” para gestionar la autenticación y la autorización. Estos son, en algunos aspectos, parecidos a las cuentas normales de estilo Unix, pero Postgres no distingue entre los usuarios y los grupos, y en su lugar prefiere el término más flexible de “rol”.

Tras la instalación, Postgres se configura para usar la autenticación ident. Esto significa que asocia los roles de Postgres con una cuenta de sistema Unix o Linux correspondiente. Si existe un rol dentro de Postgres, un nombre de usuario de Unix o Linux con el mismo nombre puede iniciar sesión ocupando ese rol.

El procedimiento de instalación creó una cuenta de usuario llamada **postgres**, que se asocia con el rol predeterminado de Postgres. Para usar Postgres, puede iniciar sesión en esa cuenta.

Existen algunas maneras de usar esta cuenta para acceder a Postgres.

### **Cambiar a la cuenta de postgres**

Cambie a la cuenta de **postgres** en su servidor escribiendo lo siguiente:

sudo -i -u postgres

Copy

Ahora, puede acceder de inmediato a una línea de comandos de PostgresSQL al escribir lo siguiente:

psql

Copy

Allí, puede interactuar con el sistema de administración de bases de datos como sea necesario.

Salga de la línea de comandos de PostgreSQL escribiendo lo siguiente:

* \q

Copy

Con esto, regresará a la línea de comandos de Linux de postgres.

### **Acceder a una línea de comandos de Postgres sin cambiar de cuenta**

También puede ejecutar el comando que desee con la cuenta de **postgres** de forma directa a través de sudo.

Por ejemplo, en el último caso se le indicó acceder a la línea de comandos de Postgres pasando primero al usuario de **postgres** y luego ejecutando psql para abrir la línea de comandos de Postgres. Puede realizarlo en un solo paso ejecutando el comando único psql como usuario de **postgres** con sudo, como se muestra:

sudo -u postgres psql

Copy

Esto le permitirá iniciar sesión de forma directa en Postgres sin el shell bash intermediario entre ellos.

De nuevo, puede salir de la sesión interactiva de Postgres escribiendo lo siguiente:

* \q

Copy

Para muchos casos de uso se requiere más de un rol de postgres. Continúe leyendo para saber cómo configurar estos roles.

## Paso 3: Crear un nuevo rol

En este momento, solo tiene el rol de **postgres** configurado dentro de la base de datos. Puede crear nuevos roles desde la línea de comandos con el comando createrole. El indicador --interactive le solicitará el nombre del nuevo rol y también le preguntará si debería tener permisos de superusuario.

Si inició sesión a través de la cuenta de **postgres**, puede crear un nuevo usuario escribiendo lo siguiente:

createuser --interactive

Copy

Si, como alternativa, prefiere usar sudo para cada comando sin dejar de usar su cuenta normal, escriba lo siguiente:

sudo -u postgres createuser --interactive

Copy

La secuencia de comandos le mostrará algunas opciones y, según sus respuestas, ejecutará los comandos correctos de Postgres para crear un usuario conforme a sus especificaciones.

Output

Enter name of role to add: sammy

Shall the new role be a superuser? (y/n) y

Puede obtener un mayor control pasando algunos indicadores adicionales. Consulte las opciones visitando la página de man:

man createuser

Copy

Ahora su instalación de Postgres tiene un usuario nuevo, pero aún no agregó bases de datos. En la sección siguiente se describe este proceso.

## Paso 4: Crear una nueva base de datos

Otra suposición que el sistema de autenticación de Postgres realiza por defecto es que para cualquier rol utilizado en el inicio de sesión habrá una base de datos con el mismo nombre al que este podrá acceder.

Esto significa que, si el usuario que creó en la última sección se llama **sammy**, ese rol intentará conectarse con una base de datos que, por defecto, también se llama “sammy”. Puede crear la base de datos apropiada con el comando createdb.

Si inició sesión a través de la cuenta de **postgres**, escribiría algo similar a lo siguiente:

* createdb Sammy

Copy

Si, como alternativa, prefiere utilizar sudo para cada comando sin dejar de emplear su cuenta normal, escribiría lo siguiente:

* sudo -u postgres createdb sammy

Copy

Esta flexibilidad ofrece varias vías para crear bases de datos cuando sea necesario.

## Paso 5: Abrir una línea de comandos de Postgres con el nuevo rol

Para iniciar sesión con la autenticación basada en ident, necesitará un usuario de Linux con el mismo nombre de su rol y su base de datos de Postgres.

Si no tiene un usuario disponible de Linux que coincida, puede crear uno con el comando adduser. Deberá hacerlo desde su cuenta non-**root** con privilegios sudo (es decir, sin iniciar sesión como usuario de **postgres**):

* sudo adduser sammy

Copy

Una vez que esté disponible esta cuenta nueva, podrá cambiar y conectarse a la base de datos escribiendo lo siguiente:

* sudo -i -u sammy
* psql

Copy

También podrá hacerlo de forma directa:

* sudo -u sammy psql

Copy

Este comando le permitirá iniciar sesión de forma automática, suponiendo que todos los componentes se hayan configurado de forma correcta.

Si desea que su usuario se conecte a una base de datos diferente, puede lograrlo especificando la base de datos de esta manera:

* psql -d postgres

Copy

Ya que inició sesión, puede verificar la información de su conexión actual escribiendo lo siguiente:

* \conninfo

Copy

Output

You are connected to database "sammy" as user "sammy" via socket in "/var/run/postgresql" at port "5432".

Esto resultará útil si se conecta a bases de datos no predeterminadas o con usuarios no predeterminados.

## Paso 6: Crear y eliminar tablas

Ahora que sabe cómo conectarse al sistema de bases de datos de PostgreSQL, puede aprender algunas tareas básicas de administración de Postgres.

La sintaxis básica para la creación de tablas es la siguiente:

CREATE TABLE table\_name (

column\_name1 col\_type (field\_length) column\_constraints,

column\_name2 col\_type (field\_length),

column\_name3 col\_type (field\_length)

);

Como puede observar, estos comandos otorgan un nombre a la tabla y luego definen las columnas, el tipo de columna y la extensión máxima de los datos de campo. De manera opcional, también puede añadir restricciones de tabla para cada columna.

Podrá obtener más información sobre [cómo crear y administrar tablas en Postgres](https://digitalocean.com/community/articles/how-to-create-remove-manage-tables-in-postgresql-on-a-cloud-server) aquí.

Para fines de demostración, cree la siguiente tabla:

* CREATE TABLE playground (
* equip\_id serial PRIMARY KEY,
* type varchar (50) NOT NULL,
* color varchar (25) NOT NULL,
* location varchar(25) check (location in ('north', 'south', 'west', 'east', 'northeast', 'southeast', 'southwest', 'northwest')),
* install\_date date
* );

Copy

Este comando creará una tabla que realiza un inventario de equipos para áreas recreativas. La primera columna de la tabla contendrá números de ID de equipos de tipo serial, que es un entero que se incrementa de forma automática. Esta columna también tiene la limitación de PRIMARY KEY, lo que significa que los valores en su interior deben ser únicos y no pueden ser nulos.

Las siguientes dos líneas crean columnas para type y color de los equipos respectivamente, que no pueden estar vacías. La siguiente línea crea una columna location y una restricción que requiere que el valor sea uno de los ocho valores posibles. La última línea crea una columna de fecha en la cual se registra la fecha en la que usted instaló el equipo.

Para dos de las columnas (equip\_id e install\_date), el comando no especifica una longitud de campo. Esto se debe a que algunos tipos de datos no requieren una longitud establecida, dado que la extensión o el formato están implícitos.

Puede ver su tabla nueva escribiendo lo siguiente:

* \d

Copy

Output

List of relations

Schema | Name | Type | Owner

--------+-------------------------+----------+-------

public | playground | table | sammy

public | playground\_equip\_id\_seq | sequence | sammy

(2 rows)

Su tabla de áreas de recreación se encuentra aquí, pero también existe algo llamado playground\_equip\_id\_seq que responde al tipo sequence. Esto es una representación del tipo serial que usted atribuyó a su columna de equip\_id. Esto realiza un seguimiento del número que sigue en la secuencia y se genera de forma automática para columnas de este tipo.

Si desea ver solo la tabla sin la secuencia, puede escribir lo siguiente:

* \dt

Copy

Output

List of relations

Schema | Name | Type | Owner

--------+------------+-------+-------

public | playground | table | sammy

(1 row)

Ahora que tenemos una tabla lista, la usaremos para practicar la administración de datos.

## Paso 7: Agregar, consultar y eliminar datos en una tabla

Ahora que tiene una tabla, puede insertar algunos datos en ella. Como ejemplo, añada un tobogán y un columpio al invocar la tabla en la que los desea agregar, nombrar las columnas y, luego, proporcionar datos para cada una de ellas de la siguiente forma:

* INSERT INTO playground (type, color, location, install\_date) VALUES ('slide', 'blue', 'south', '2017-04-28');
* INSERT INTO playground (type, color, location, install\_date) VALUES ('swing', 'yellow', 'northwest', '2018-08-16');

Copy

Debe tener cuidado al ingresar los datos para evitar algunos errores comunes. Para empezar, no escriba los nombres de las columnas entre comillas. Estás sí se necesitarán para los valores de la columna que ingresó.

Otro aspecto que debe tener en cuenta es no ingresar un valor para la columna equip\_id. Esto se debe a que se genera de forma automática siempre que se añade una fila nueva a la tabla.

Recupere la información que agregó escribiendo lo siguiente:

* SELECT \* FROM playground;

Copy

Output

equip\_id | type | color | location | install\_date

----------+-------+--------+-----------+--------------

1 | slide | blue | south | 2017-04-28

2 | swing | yellow | northwest | 2018-08-16

(2 rows)

Aquí, puede ver que su equip\_id se completó con éxito y que todos sus otros datos se organizaron de forma correcta.

Si el tobogán del área de recreación se daña y tiene que eliminarlo, también puede eliminar la fila de su tabla escribiendo lo siguiente:

* DELETE FROM playground WHERE type = 'slide';

Copy

Consulte la tabla de nuevo:

* SELECT \* FROM playground;

Copy

Output

equip\_id | type | color | location | install\_date

----------+-------+--------+-----------+--------------

2 | swing | yellow | northwest | 2018-08-16

(1 row)

Observará que la fila de slide ya no se encuentra en la tabla.

## Paso 8: Agregar y eliminar columnas en una tabla

Después de crear una tabla, puede modificarla añadiendo o eliminando columnas. Agregue una columna para mostrar la última visita de mantenimiento por cada equipo escribiendo lo siguiente:

* ALTER TABLE playground ADD last\_maint date;

Copy

Si vuelve a ver la información de su tabla, observará que se agregó la columna nueva, pero no se ingresaron datos:

* SELECT \* FROM playground;

Copy

Output

equip\_id | type | color | location | install\_date | last\_maint

----------+-------+--------+-----------+--------------+------------

2 | swing | yellow | northwest | 2018-08-16 |

(1 row)

Si determina que su equipo de trabajo utiliza una herramienta separada para dar seguimiento al historial de mantenimiento, puede eliminar la columna escribiendo lo siguiente:

* ALTER TABLE playground DROP last\_maint;

Copy

Con esto, se eliminan la columna last\_maint y los valores que se encuentren en ella, pero deja intactos todos los demás datos.

## Paso 9: Actualizar datos de una tabla

Hasta ahora, a través de este tutorial aprendió a agregar registros a una tabla y a eliminarlos de ella, pero aún no se abordó la forma de modificar los registros existentes.

Puede actualizar los valores de una entrada existente buscando el registro que desee y fijando el valor que prefiera utilizar para la columna. Puede consultar el registro de swing (se incluirán todos los columpios de su tabla) y cambiar su color a red. Esto puede ser útil si asignó una tarea de pintura al columpio:

* UPDATE playground SET color = 'red' WHERE type = 'swing';

Copy

Puede verificar la eficacia de la operación consultando los datos de nuevo:

* SELECT \* FROM playground;

Copy

Output

equip\_id | type | color | location | install\_date

----------+-------+-------+-----------+--------------

2 | swing | red | northwest | 2018-08-16

(1 row)

Como puede ver, ahora, el columpio está registrado como rojo.

**PGSQL**

<https://medium.com/coding-blocks/creating-user-database-and-adding-access-on-postgresql-8bfcd2f4a91e>

**SQL SERVER**

<https://computingforgeeks.com/how-to-install-ms-sql-on-ubuntu/>

<https://docs.microsoft.com/en-us/sql/connect/php/installation-tutorial-linux-mac?view=sql-server-ver15>