

L3

Vamos a ir instalando los componentes de LAMP en Centos y Ubuntu

-Lo primero que vamos a hacer es instalar Apache

En centos tenemos que poner yum install httpd, estando en su.

*Importante hacer ifup de las tarjetas para poder descargar, para mirar el nombre de las tarjetas ponemos el comando: ip address

Si queremos ver el estado en el que se crea el servicio httpd ponemos el comando: systemctl status httpd y veremos que está cargado pero no activo.

Para habilitarlo, hacemos el comando: systemctl enable httpd, que crea un enlace simbolico, colocado en un sitio para que el SO sepa que lo tiene que arrancar cada vez que se encienda la maquina

Podemos comprobar con el comando: systemctl status httpd, que aún no hemos activado el servicio, asique tenemos que arrancarlo utilizando el comando: systemctl start httpd. Ahora podemos volver a probar el comando systemctl status httpd y vemos que nos aparece activo.

La ubicación de la instalacion está en /etc/httpd

Para comprobar que está funcionando vamos a copiar la direccion ip de nuestra maquina centos en internet, en mi caso es la 192.168.56.110, y vemos que no nos deja acceder.

Ahora comprobamos con curl, para ver si podemos acceder y usamos el comando: curl https://localhost, vemos que este comando si nos devuelve el código de una página, por lo que vemos, que internamente, desde nuestra máquina si podemos acceder, pero desde fuera (utilizando el navegador) no podemos.

Vamos a probarlo ahora desde la terminal del host, con el comando: curl http://192.168.56.110 y vemos que nos devuelve que la conexión ha sido rechazada, por lo que ponemos entender que tenemos un firewall lanzado.

Como pensamos que tenemos un firewall activo, vamos a utilizar el comando: firewall-cmd --state y vemos que nos dice que el firewall esta funcionando.

Miramos las reglas que tenemos establecidas en el firewall con el comando: firewall-cmd --list-all y con el comando: firewall-cmd --list-all-zones nos aparecera la lista de zonas que tenemos configurada, con los distintos comportamientos que tiene que tener nuestro firewall segun la zona que tengamos habilitada, por defecto tenemos la zona pública. Con esto podemos definir una serie de reglas para el firewall dependiendo del entorno en el que usamos el buscador.

Para dejar que pueda acceder con el navegador, tenemos que añadir el puerto 80 o el servicio http en la zona publica, que es la que hemos dicho

que está activa por defecto, esto lo hacemos con el comando: `firewall-cmd --zone=public --add-service=http`, pero si lo hacemos de esta forma, esta regla no quedará guardada y al apagar el ordenador se perderá, por esto, añadimos a la línea anterior la opción: `--permanent` quedando entonces el comando: `firewall-cmd --zone=public --permanent --add-service=http`. Si lo quisiéramos hacer con el puerto cambiaríamos la última opción del comando poniendo `--add-port=80`.

Esto que hemos hecho escribe de manera permanente el archivo de configuración del cortafuegos de la zona pública, pero aun así tenemos que decirle que recargue el archivo para que cargue los cambios que hemos hecho. Esto lo hacemos con el comando: `firewall-cmd --reload` y ahora podemos consultar el firewall con el comando: `firewall-cmd --list-all` y veremos que se ha añadido la regla que hemos puesto.

Una vez hecho esto probamos de nuevo en el navegador a buscar la IP de nuestra máquina CentOS y vemos que ahora sí encuentra la página y muestra información.

En Ubuntu tenemos que poner `sudo apt install apache2`.

Si queremos ver el estado en el que se crea el servicio `apache2` ponemos el comando: `systemctl status apache2`, en este caso, a mí me sale cargado y activo.

Hay que seguir los mismos pasos que en CentOS, pero yo por lo menos he tenido que instalar la `firewall-cmd`, una vez instalado, los pasos y las instrucciones son iguales que en CentOS.

-Ahora vamos a instalar php

-En CentOS tenemos que instalarlo con el comando: `yum install php`
La configuración de php está en el directorio `/etc/php.ini`

-En Ubuntu tenemos que instalarlo con el comando: `sudo apt install php`

Con php podemos limitar el tamaño de archivos de subida, para que no colapse nuestro servidor definir modo depuración... Por defecto envía los errores que se producen al mismo lugar que apache. Todos los logs van a `/var/log`

Si nos metemos en `/var/log/httpd`, con el comando: `cd /var/log/httpd` podemos ver dos archivos, `access_log` y `error_log`, estos archivos que guardan los logs no suelen guardarse durante mucho tiempo, ya que pueden ocupar mucho espacio. Los errores que se generen cuando interpretemos una página web con php se irán al archivo `error_log`

Ahora vamos a acceder a la raíz de documentos, lo hacemos con el comando: `cd /var/www/html` está en `var`, porque en este directorio se almacenan todos los datos variables, los datos de usuario

En este directorio es donde vamos a posicionar las páginas web, donde tenemos que publicar el contenido php que vamos a ejecutar en el final de la práctica.

En el siguiente paso vamos a utilizar nano, asique en centos lo instalamos con el comando: yum install nano

Vamos a crear una pagina de prueba, para esto utilizamos el comando: nano -w index.html que es el nombre del fichero por defecto que apache va a buscar, esto tambien lo podemos configurar dentro del fichero httpd.conf esta indicado expresamente que se utiliza el archivo index.html

Dentro de este archivo metemos el siguiente código:

```
<html>
<head><title>Hola</title></head>
<body>
<H1> Probando apache </h1>
</body>
</html>
```

Y lo probamos accediendo a el con el navegador poniendo la IP de nuestra maquina Centos, esta vez en el buscador debe aparecernos la linea Probando apache, que es la que hemos escrito en el fichero index.html

Ahora vamos a crear un archivo php, que se llamará hola, y vamos a copiar en el lo que ya hemos escrito en index.html, lo hacemos con el comando: cp index.html hola.php, y ahora modificamos hola.php con el comando: nano -w hola.php y lo dejamos de la siguiente forma:

```
<html>
<head><title>Hola</title></head>
<body>
<H1> Probando PHP </h1>
phpinfo();
?>php
    phpinfo();
?>
</body>
</body>
</html>
```

Y lo guardamos. Utilizamos phpinfo() porque es una función de php que nos da la configuración de php en el sistema local, se utiliza para hacer alguna depuración.

Ahora volvemos a ir al navegador y escribimos la dirección de nuestra maquina centos seguido de /hola.php, en mi caso queda 192.168.56.110/hola.php

Al darle a enter vemos que nos muestra en nuevo mensaje de Probando PHP, pero la informacion de phpinfo no aparece. Si damos click derecho sobre el buscador e inspeccionamos elemento, vemos que la linea de la función phpinfo aparece comentada, esto significa que realmente no tenemos php activo para apache.

Para solucionar esto tenemos que reiniciar apache, lo podemos hacer con el comando: apachectl restart.

-Ahora vamos a instalar MySQL

Como no es de código 100% abierto, vamos a instalar MariaDB

-En CentOS lo hacemos con el comando: `yum install mariadb-server`

-En Ubuntu lo hacemos con el comando: `apt install mariadb-server`

Vemos el estado con el comando: `systemctl status mariadb` y vemos que está cargado pero no está activo.

Como hemos hecho antes con Apache, lo habilitamos con el comando: `systemctl enable mariadb-server`, en Ubuntu no hace falta hacerlo.

Para probar que funciona usamos el comando: `mysql mysql` donde el segundo `mysql` hace referencia a la base de datos que queremos cargar, en este caso la tabla por defecto. Con esto habríamos demostrado que está funcionando y que me puedo conectar.

Estando dentro de `mysql`, podemos lanzar el comando: `show tables;` y muestra las tablas que hay en la base de datos `mysql`.

Ahora vamos a conectar desde PHP hacia `mysql`. Para esto nos desplazamos al directorio `/var/www/html` con el comando: `cd /var/www/html` y abrimos el archivo `hola.php` con el comando: `nano -w hola.php` lo modificamos y debe de quedar así.

```
<html>
<head><title>Hola</title></head>
<body>
<h1> Probando PHP </h1>
<h2>Conexión Mysql </h2>
<?php
    $mysql = new mysqli('localhost', 'usuario', 'secreto','ise2020');
    if ($mysql->connect_error) {
        die('No pudo conectarse: ' . $mysql->connect_error);
    }
    echo '<h1>Conectado satisfactoriamente</h1>';
?>
<h2>Configuración de PHP</h2>
<?php
    phpinfo();
?>
</body>
</html>
```

El código que hemos añadido, busca básicamente abrir una nueva conexión `mysql` en el ordenador local con un usuario, un secreto (una contraseña) y una base de datos `ise2020`. Para esto tenemos que crear un usuario, un secreto y una base de datos.

Abrimos otra terminal, conectada por `ssh` a nuestra máquina CentOS o Ubuntu, según con cuál la estemos haciendo, lanzamos el comando: `mysql` y dentro de `mysql` lanzamos el comando: `create database ise2020;` para crear la base de datos.

Vamos a crear ahora un usuario que tenga derechos de acceso, seguimos en `mysql`,

usamos ahora el comando: `grant all privileges on ise2020.* to 'alumno'@'localhost' identified by '123456';` con este comando le decimos que al usuario alumno conectado desde localhost y con contraseña 123456 le de todos los privilegios en la base de datos

Vamos a probar ahora que lo que hemos hecho funciona, lanzamos el comando: `mysql -ualumno -p123456 ise2020` y vemos que accede bien a la base de datos

Ahora nos dirigimos al otro terminal en el que tenemos abierto el archivo hola.php y cambiamos los parámetros de usuario y secreto, por los que hemos dado al crear el usuario en la base de datos, en mi caso usuario cambia a alumno y secreto a 123456 y nos queda el código así:

```
<html>
<head><title>Hola</title></head>
<body>
<H1> Probando PHP </h1>
<h2>Conexin Mysql </h2>
<?php
    $mysql = new mysqli('localhost', 'alumno', '123456','ise2020');
    if ($mysql->connect_error) {
        die('No pudo conectarse: ' . $mysql->connected_error);
    }
    echo '<h1>Conectado satisfactoriamente</h1>';
?>
<h2>Configuración de PHP</h2>
<?php
    phpinfo();
?>
</body>
</html>
```

Pero si recargamos la pagina en el navegador, como hemos hecho antes poniendo la ip de la maquina centos/hola.php va a mostrarnos el texto pero no va a funcionar la linea de código "`$mysql = new mysqli...`" esto es porque falta instalar la libreria de mysql en php. Para solucionar esto usamos el comando: `yum install php-mysql`.

Una vez hecho esto tenemos que recargar de nuevo apache ya que el interprete esta corriendo en el espacio de memoria de apache, lo hacemos con el comando: `apachectl restart`

Volvemos al buscador y recargamos la pagina y ahora si nos aparece el mensaje de conectado satisfactoriamente.

Para hacer el ejercicio pendiente que dejo el profesor tenemos que usar la base de datos ise2020 que ya hemos creado, para esto, estando en mysql lanzamos el comando: `show databases;` y vemos que la que hemos creado aparece, para empezar a operar en ella, usamos el comando: `use ise2020`, con lo que nos cambiamos a la base de datos ise2020.

Ahora necesitamos crear la tabla que se nos pide, por lo que lanzamos el comando: `create table alumnos (id INT PRIMARY KEY, nombre VARCHAR(20), apellidos VARCHAR(30), email VARCHAR(30));`

Y para darle algunos valores vamos a usar el comando: INSERT INTO alumnos VALUES(1, 'Manuel', 'Cano', 'mjcr@correo.ugr.es'); este representaría el primer individuo guardado en la tabla, creamos un par más cambiando los valores.

Vamos al archivo hola.php y añadimos el código de al final del documento de auto aprendizaje de la lección 3 se debe quedar así:

```
<html>
<head><title>Hola</title></head>
<body>
<H1> Probando PHP </h1>
<h2>Conexin Mysql </h2>
<?php
    $mysql = new mysqli('localhost', 'alumno', '123456','ise2020');
    if ($mysql->connect_error) {
        die('No pudo conectarse: ' . $mysql->connected_error);
    }
    echo '<h1>Conectado satisfactoriamente</h1>';
?>
<h1>Contenido de la tabla Alumnos</h1>
<table border="1">
<tr><td>Id</td><td>Nombre</td><td>Email</td></tr>
<?php
    $sql = "SELECT id, nombre, apellidos, email FROM alumnos";
    $result = $mysql->query($sql);

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            echo "<tr><td>".$row["id"]."</td><td>".$row["nombre"]." ".
                $row["apellidos"]."</td>";
            echo "<td>".$row["email"]."</td></tr>\n";
        }
    }

    mysql_close($enlace);
    phpinfo();
?>
<h2>Configuración de PHP</h2>
</body>
</html>
```

Guardamos el archivo

Reiniciamos el apache con el comando: apachectl restart, nos vamos a nuestro navegador y volvemos a poner la direccion de nuestro centos/hola.php en mi caso se queda así: <http://192.168.56.110/hola.php> y al darle a enter debe de salir una pequeña tabla con las entradas que hemos insertado.