**Cómo instalar la pila Linux, Apache, MySQL y PHP (LAMP) en Ubuntu 20.04**

Published on May 19, 2020

* [Ubuntu 20.04](https://www.digitalocean.com/community/tags/ubuntu-20-04)
* [LAMP Stack](https://www.digitalocean.com/community/tags/lamp-stack)
* [PHP](https://www.digitalocean.com/community/tags/php)
* [Apache](https://www.digitalocean.com/community/tags/apache)
* [MySQL](https://www.digitalocean.com/community/tags/mysql)
* [Ubuntu](https://www.digitalocean.com/community/tags/ubuntu)



By

Español



**Introducción**

Una pila “LAMP” es un conjunto de aplicaciones de software de código abierto que se suelen instalar juntas para que un servidor pueda alojar aplicaciones y sitios web dinámicos escritos en PHP. Este término es en realidad un acrónimo que representa al sistema operativo **L**inux, con el servidor web **A**pache. Los datos del sitio se almacenan en una base de datos **M**ySQL y el contenido dinámico se procesa mediante **P**HP.

En esta guía, instalaremos una pila LAMP en un servidor Ubuntu 20.04.

**Requisitos previos**

Para completar este tutorial, deberá disponer de un servidor de Ubuntu 20.04 con una cuenta sudo non-root user y un firewall básico. Puede realizar esta configuración siguiendo nuestra [Guía de configuración inicial de servidores para Ubuntu 20.04](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-20-04).

**Paso 1: Instalar Apache y actualizar el firewall**

El servidor web Apache está entre los más populares del mundo. Está bien documentado, tiene una comunidad de usuarios activa y ha sido muy utilizado durante gran parte de la historia de la web, por lo que es una excelente opción predeterminada para alojar sitios web.

Instale Apache usando el administrador de paquetes de Ubuntu, apt:

1. sudo apt update
2. sudo apt install apache2

Si es la primera vez que utiliza sudo en esta sesión, se le pedirá que proporcione su contraseña de usuario para confirmar que tenga los privilegios adecuados para administrar los paquetes del sistema con apt.  También se le solicitará que confirme la instalación de Apache al pulsar Y y ENTER.

Una vez que la instalación se complete, deberá ajustar la configuración de su firewall para permitir tráfico HTTP y HTTPS. UFW tiene diferentes perfiles de aplicaciones que puede aprovechar para hacerlo. Para enumerar todos los perfiles de aplicaciones de UFW disponibles, puede ejecutar lo siguiente:

1. sudo ufw app list

Verá un resultado como este:

Output

Available applications:

Apache

Apache Full

Apache Secure

OpenSSH

A continuación, explicamos cada uno de estos perfiles:

* **Apache**: este perfil abre solo el puerto 80 (tráfico web normal no cifrado).
* **Apache Full**: este perfil abre los puertos 80 (tráfico web normal no cifrado) y 443 (tráfico TLS/SSL cifrado).
* **Apache Secure**: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado).

Por ahora, es mejor permitir conexiones únicamente en el puerto 80, ya que se trata de una instalación nueva de Apache y todavía no tiene un certificado TLS/SSL configurado para permitir tráfico HTTPS en su servidor.

Para permitir tráfico únicamente en el puerto 80 utilice el perfil Apache:

1. sudo ufw allow in "Apache"

Puede verificar el cambio con lo siguiente:

1. sudo ufw status

Output

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

Apache ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

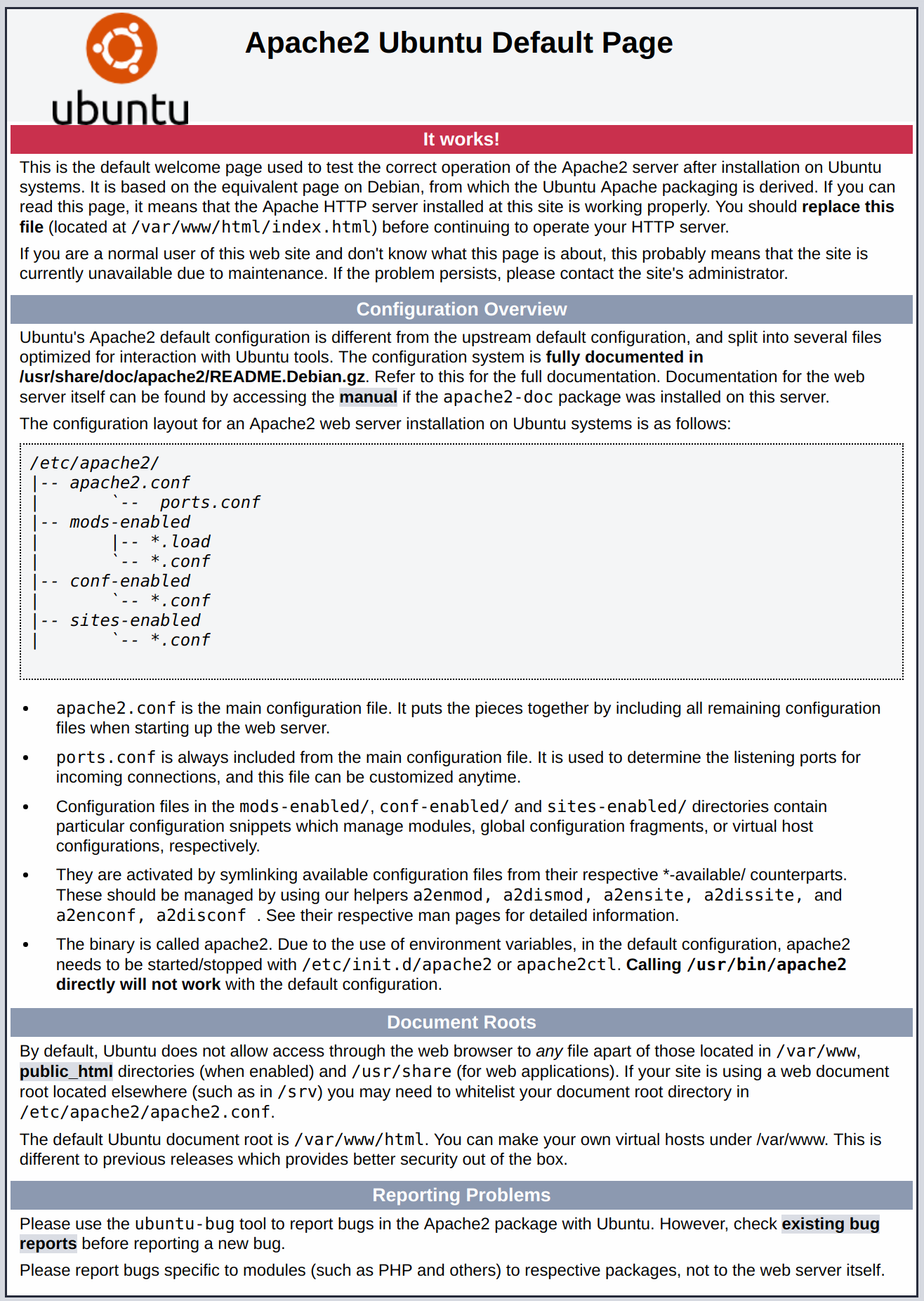
Apache (v6) ALLOW Anywhere (v6)

Ahora, se permite tráfico en el puerto 80 a través del firewall.

Puede realizar una verificación rápida para comprobar que todo se haya realizado según lo previsto dirigiéndose a la dirección IP pública de su servidor en su navegador web (consulte la nota de la siguiente sección para saber cuál es su dirección IP pública si no dispone de esta información):

http://your\_server\_ip

Verá la página web predeterminada de Apache para Ubuntu 20.04, que se encuentra allí para fines informativos y de prueba. Debería tener un aspecto similar a este:



Si ve esta página, su servidor web estará correctamente instalado y el acceso a él será posible a través de su firewall.

**Cómo averiguar la dirección IP pública de su servidor**

Si no conoce la dirección IP pública de su servidor, hay varias formas de encontrarla. Por lo general, es la dirección que utiliza para establecer conexión con su servidor a través de SSH.

Existen varias formas de hacerlo desde la línea de comandos. Primero, podría usar las herramientas de iproute2 para obtener su dirección IP escribiendo esto:

1. ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\/.\*$//'

Esto nos brindará dos o tres líneas. Todas estas direcciones son correctas, pero su computadora puede usar una de ellas. Por ello, no dude en probarlas todas.

Un método alternativo consiste en usar la utilidad curl para contactar a una parte externa a fin de que le indique *su* evaluación del servidor. Esto se hace solicitando a un servidor específico su dirección IP:

1. curl http://icanhazip.com

Independientemente del método que utilice para obtener su dirección IP, escríbala en la barra de direcciones de su navegador web para ver la página predeterminada de Apache.

**Paso 2: Instalar MySQL**

Ahora que dispone de un servidor web funcional, deberá instalar un sistema de base de datos para poder almacenar y gestionar los datos de su sitio. MySQL es un sistema de administración de bases de datos popular que se utiliza en entornos PHP.

Una vez más, utilice apt para adquirir e instalar este software:

1. sudo apt install mysql-server

Cuando se le solicite, confirme la instalación al escribir Y y, luego, ENTER.

Cuando la instalación se complete, se recomienda ejecutar una secuencia de comandos de seguridad que viene preinstalada en MySQL Con esta secuencia de comandos se eliminarán algunos ajustes predeterminados poco seguros y se bloqueará el acceso a su sistema de base de datos. Inicie la secuencia de comandos interactiva ejecutando lo siguiente:

1. sudo mysql\_secure\_installation

Se le preguntará si desea configurar el VALIDATE PASSWORD PLUGIN.

**Nota**: La habilitación de esta característica queda a discreción del usuario. Si se habilita, MySQL rechazará con un mensaje de error las contraseñas que no coincidan con los criterios especificados. Dejar la validación desactivada será una opción segura, pero siempre deberá utilizar contraseñas seguras y únicas para credenciales de bases de datos.

Elija Y para indicar que sí, o cualquier otra cosa para continuar sin la habilitación.

VALIDATE PASSWORD PLUGIN can be used to test passwords

and improve security. It checks the strength of password

and allows the users to set only those passwords which are

secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No:

Si responde “sí”, se le solicitará que seleccione un nivel de validación de contraseña. Tenga en cuenta que, si ingresa 2 para indicar el nivel más seguro, recibirá mensajes de error al intentar establecer cualquier contraseña que no contenga números, letras en mayúscula y minúscula, y caracteres especiales, o que se base en palabras comunes del diccionario.

There are three levels of password validation policy:

LOW Length >= 8

MEDIUM Length >= 8, numeric, mixed case, and special characters

STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1

Independientemente de que haya elegido instalar el VALIDATE PASSWORD PLUGIN, su servidor le solicitará, a continuación, que seleccione y confirme una contraseña para el **root** user de MySQL. No debe confundirse con el **root del sistema**. El **root user de base de datos** es un usuario administrativo con privilegios completos sobre el sistema de base de datos. Si bien el método de autenticación predeterminado del root user de MySQL no requiere el uso de una contraseña, **incluso si hay una establecida**, deberá definir una contraseña segura en este punto como una medida de seguridad adicional. Hablaremos de esto en breve.

Si habilitó la validación de contraseña, se le indicará la seguridad de la contraseña del root user que acaba de ingresar y su servidor le preguntará si desea continuar usándola. Si está conforme con su contraseña actual, ingrese Y para indicar “sí” en la solicitud:

Estimated strength of the password: 100

Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y

Para el resto de las preguntas, presione Y y ENTER en cada mensaje. 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 MySQL aplique de inmediato los cambios que realizó.

Cuando termine, compruebe si puede iniciar sesión en la consola de MySQL al escribir lo siguiente:

1. sudo mysql

Esto permitirá establecer conexión con el servidor de MySQL como **root** user de la base de datos administrativa, lo que se infiere del uso de sudo cuando se ejecuta este comando. Debería ver el siguiente resultado:

Output

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 22

Server version: 8.0.19-0ubuntu5 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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>

Para salir de la consola de MySQL, escriba lo siguiente:

1. exit

Observe que no necesitó proporcionar una contraseña para conectarse como **root** user, a pesar de que definió una al ejecutar la secuencia de comandos mysql\_secure\_installation. Esto se debe a que el método de autenticación predeterminado para el usuario administrativo de MySQL es unix\_socket en vez de password. Si bien esto puede parecer un problema de seguridad inicialmente, hace que el servidor de la base de datos sea más seguro porque los únicos usuarios que pueden iniciar sesión como **root** user de My SQL son los usuarios del sistema con privilegios sudo que establecen conexión desde la consola o mediante una aplicación que se ejecute con los mismos privilegios. En términos prácticos, eso significa que no podrá usar el usuario **root** de la base de datos administrativa para establecer conexión desde su aplicación PHP. Establecer una contraseña para la cuenta **root** de MySQL es una medida de protección, en caso de que el método de autenticación predeterminado se cambie de unix\_socket a password.

Para mayor seguridad, es mejor disponer de cuentas de usuario dedicadas con privilegios de menor alcance configurados para cada base de datos, en especial si planea disponer de varias bases de datos alojadas en su servidor.

**Nota:** Al momento de la redacción de este artículo, la biblioteca PHP nativa de MySQL mysqlnd [no admite](https://www.php.net/manual/en/ref.pdo-mysql.php) caching\_sha2\_authentication, el método de autenticación predeterminado de MySQL 8. Por este motivo, al crear usuarios de bases de datos para aplicaciones PHP en MySQL 8, deberá asegurarse de que estén configurados para usar mysql\_native\_password en su lugar. Demostraremos cómo hacerlo en el [paso 6](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-es#step-6-%E2%80%94-testing-database-connection-from-php-(optional)).

Ahora, su servidor de MySQL está instalado y protegido. A continuación, instalaremos PHP, el componente final de la pila LAMP.

**Paso 3: Instalar PHP**

Instaló Apache para presentar su contenido y MySQL para almacenar y gestionar sus datos. PHP es el componente de nuestra configuración que procesará el código para mostrar contenido dinámico al usuario final. Además del paquete php, necesitará php-mysql, un módulo PHP que permite que este se comunique con bases de datos basadas en MySQL. También necesitará libapache2-mod-php para habilitar Apache para gestionar archivos PHP. Los paquetes PHP básicos se instalarán automáticamente como dependencias.

Para instalar estos paquetes, ejecute lo siguiente:

1. sudo apt install php libapache2-mod-php php-mysql

Una vez que la instalación se complete, podrá ejecutar el siguiente comando para confirmar su versión de PHP:

1. php -v

Output

PHP 7.4.3 (cli) (built: Mar 26 2020 20:24:23) ( NTS )

Copyright (c) The PHP Group

Zend Engine v3.4.0, Copyright (c) Zend Technologies

with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies

En este punto, su pila LAMP está plenamente operativa, pero, para poder probar su configuración con una secuencia de comandos PHP, lo mejor es instalar un [host virtual de Apache](https://httpd.apache.org/docs/current/vhosts/) adecuado para almacenar los archivos y las carpetas de su sitio web. Lo haremos en el siguiente paso.

**Paso 4: Crear un host virtual para su sitio web**

Al emplear el servidor web Apache, puede crear *hosts virtuales* (similares a los bloques de servidor en Nginx) para encapsular detalles de configuración y alojar más de un dominio desde un único servidor. En esta guía, configurará un dominio llamado **your\_domain**, **pero deberá cambiar este nombre por el de su dominio propio**.

**Nota**: En caso de que utilice DigitalOcean como proveedor de hosting de DNS, puede consultar nuestros [documentos sobre productos](https://www.digitalocean.com/docs/networking/dns/how-to/manage-records/#a-records) para hallar instrucciones detalladas sobre cómo configurar un nuevo nombre de dominio y orientarlo hacia su servidor.

Ubuntu 20.04 tiene habilitado un bloque de servidor por defecto, que está configurado para proporcionar documentos del directorio /var/www/html. Si bien esto funciona bien para un solo sitio, puede ser difícil de manejar si aloja varios. En lugar de modificar /var/www/html, crearemos una estructura de directorio dentro de /var/www para el sitio **your\_domain** y dejaremos /var/www/html establecido como directorio predeterminado que se presentará si una solicitud de cliente no coincide con ningún otro sitio.

Cree el directorio para **your\_domain** de la siguiente manera:

1. sudo mkdir /var/www/your\_domain

A continuación, asigne la propiedad del directorio con la variable de entorno $USER, que hará referencia a su usuario de sistema actual:

1. sudo chown -R $USER:$USER /var/www/your\_domain

Luego, abra un nuevo archivo de configuración en el directorio sites-available de Apache usando el editor de línea de comandos que prefiera. En este caso, utilizaremos nano:

1. sudo nano /etc/apache2/sites-available/your\_domain.conf

De esta manera, se creará un nuevo archivo en blanco. Pegue la siguiente configuración básica:

/etc/apache2/sites-available/your\_domain.conf

<VirtualHost \*:80>

ServerName your\_domain

ServerAlias www.your\_domain

ServerAdmin webmaster@localhost

DocumentRoot /var/www/your\_domain

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Con esta configuración de VirtualHost, le indicamos a Apache que proporcione your\_domain usando /var/www/your\_domain como directorio root web. Si desea probar Apache sin un nombre de dominio, puede eliminar o convertir en comentario las opciones ServerName y ServerAlias añadiendo un carácter # al principio de las líneas de cada opción.

Ahora, puede usar a2ensite para habilitar el nuevo host virtual:

1. sudo a2ensite your\_domain

Puede ser conveniente deshabilitar el sitio web predeterminado que viene instalado con Apache. Es necesario hacerlo si no se utiliza un nombre de dominio personalizado, dado que, en este caso, la configuración predeterminada de Apache sobrescribirá su host virtual. Para deshabilitar el sitio web predeterminado de Apache, escriba lo siguiente:

1. sudo a2dissite 000-default

Para asegurarse de que su archivo de configuración no contenga errores de sintaxis, ejecute lo siguiente:

1. sudo apache2ctl configtest

Por último, vuelva a cargar Apache para que estos cambios surtan efecto:

1. sudo systemctl reload apache2

Ahora, su nuevo sitio web está activo, pero el directorio root web /var/www/your\_domain todavía está vacío. Cree un archivo index.html en esa ubicación para poder probar que el host virtual funcione según lo previsto:

1. nano /var/www/your\_domain/index.html

Incluya el siguiente contenido en este archivo:

/var/www/your\_domain/index.html

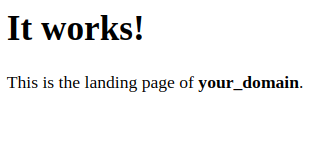
<h1>It works!</h1>

<p>This is the landing page of <strong>your\_domain</strong>.</p>

Ahora, diríjase a su navegador y acceda al nombre de dominio o la dirección IP de su servidor una vez más:

http://server\_domain\_or\_IP

Verá una página como la siguiente:



Si ve esta página, su host virtual de Apache está funcionando según lo previsto.

Puede dejar este archivo establecido como página de destino temporal de su aplicación hasta que configure un archivo index.php que lo sustituya. Cuando lo haga, recuerde eliminar el archivo index.html de su root de documentos, o cambiarle el nombre, ya que tendría precedencia sobre un archivo index.php por defecto.

**Nota sobre DirectoryIndex en Apache**

Con la configuración predeterminada de DirectoryIndex en Apache, un archivo denominado index.html siempre tendrá prioridad sobre un archivo index.php. Esto es útil para establecer páginas de mantenimiento en aplicaciones PHP, dado que se puede crear un archivo index.html temporal que contenga un mensaje informativo para los visitantes. Como esta página tendrá precedencia sobre la página index.php, se convertirá en la página de destino de la aplicación. Una vez que el mantenimiento se completa, el archivo index.html se elimina del root de documentos, o se le cambia el nombre, para volver mostrar la página habitual de la aplicación.

Si desea cambiar este comportamiento, deberá editar el archivo /etc/apache2/mods-enabled/dir.conf y modificar el orden en el que el archivo index.php se enumera en la directiva DirectoryIndex:

1. sudo nano /etc/apache2/mods-enabled/dir.conf

/etc/apache2/mods-enabled/dir.conf

<IfModule mod\_dir.c>

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

Después de guardar y cerrar el archivo, deberá volver a cargar Apache para que los cambios surtan efecto:

1. sudo systemctl reload apache2

En el siguiente paso, crearemos una secuencia de comandos PHP para probar que PHP esté correctamente instalado y configurado en su servidor.

**Paso 4: Probar el procesamiento de PHP en su servidor web**

Ahora que dispone de una ubicación personalizada para alojar los archivos y las carpetas de su sitio web, crearemos una secuencia de comandos PHP de prueba para verificar que Apache pueda gestionar solicitudes y procesar solicitudes de archivos PHP.

Cree un archivo nuevo llamado info.php dentro de su carpeta root web personalizada:

1. nano /var/www/your\_domain/info.php

Con esto se abrirá un archivo vacío. Añada el siguiente texto, que es el código PHP válido, dentro del archivo:

/var/www/your\_domain/info.php

**<?php**

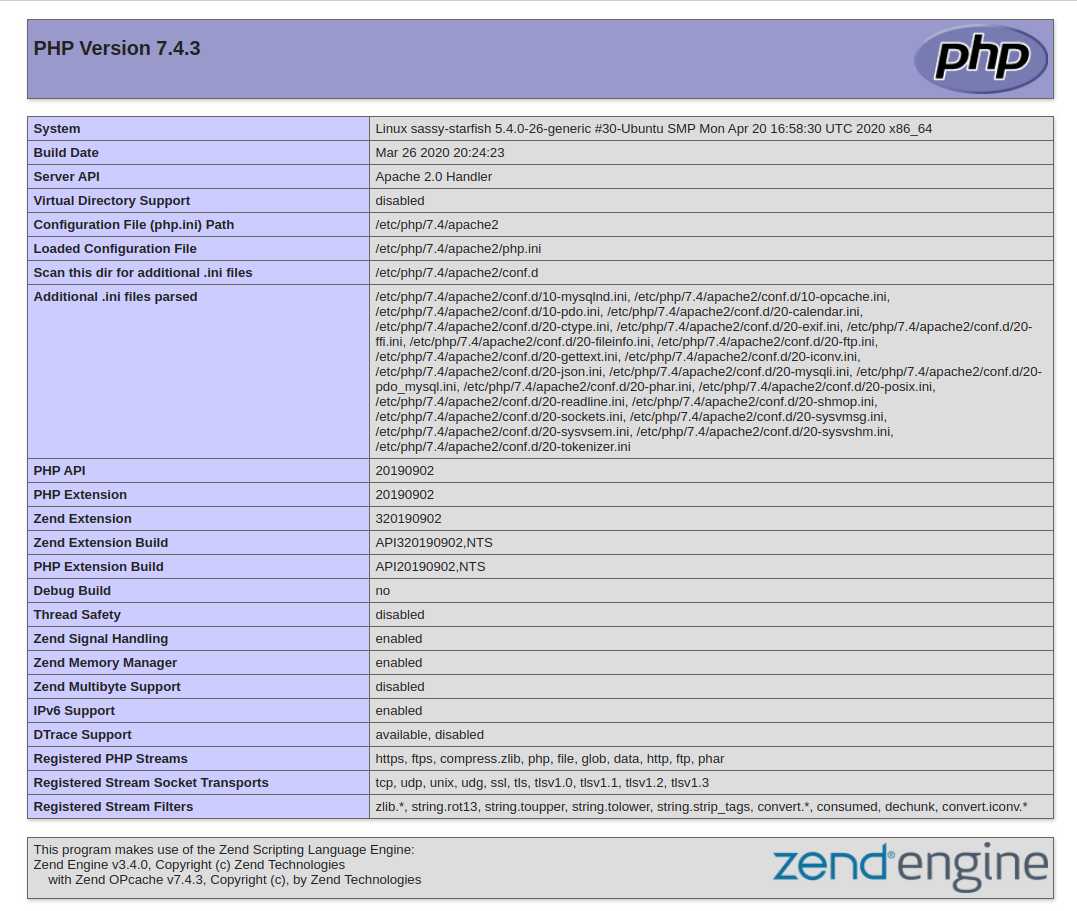
phpinfo();

Cuando termine, guarde y cierre el archivo.

Para probar esta secuencia de comandos, diríjase a su navegador web y acceda al nombre de dominio o la dirección IP de su servidor, seguido del nombre de la secuencia de comandos, que en este caso es info.php:

http://server\_domain\_or\_IP/info.php

Verá una página similar a la siguiente:



En esta página, se proporciona información básica sobre su servidor desde la perspectiva de PHP. Es útil para la depuración y para asegurarse de que sus ajustes se apliquen correctamente.

Si puede ver esta página en su navegador, su instalación de PHP funciona según lo previsto.

Tras comprobar la información pertinente sobre su servidor PHP a través de esa página, es recomendable que elimine el archivo que creó, dado que contiene información confidencial sobre su entorno PHP y su servidor de Ubuntu. Puede usar rm para hacerlo:

1. sudo rm /var/www/your\_domain/info.php

Siempre puede recrear esta página si necesita acceder a la información posteriormente.

**Paso 6: Probar la conexión con la base de datos desde PHP (opcional)**

Si desea probar si PHP puede establecer conexión con MySQL y ejecutar consultas a la base de datos, puede crear una tabla de prueba con datos ficticios y realizar consultas relacionadas con su contenido con una secuencia de comandos PHP. Para poder hacerlo, debemos crear una base de datos de prueba y un nuevo usuario de MySQL debidamente configurado para acceder a ella.

Al momento de la redacción de este artículo, la biblioteca PHP nativa de MySQL mysqlnd [no admite](https://www.php.net/manual/en/ref.pdo-mysql.php) caching\_sha2\_authentication, el método de autenticación predeterminado de MySQL 8. Vamos a tener que crear un usuario nuevo con el método de autenticación mysql\_native\_password para poder establecer conexión con la base de datos de MySQL desde PHP.

Crearemos una base de datos denominada **example\_database** y un usuario llamado **example\_user**, pero puede sustituir estos nombres por valores diferentes.

Primero, establezca conexión con la consola de MySQL usando la cuenta **root**:

1. sudo mysql

Para crear una base de datos nueva, ejecute el siguiente comando desde su consola de MySQL:

1. CREATE DATABASE example\_database;

Ahora puede crear un nuevo usuario y concederle privilegios completos sobre la base de datos personalizada que acaba de crear.

El siguiente comando crea un usuario nuevo llamado example\_user, que utiliza mysql\_native\_password como método de autenticación predeterminado. Definimos la contraseña de este usuario como password, pero debe sustituir este valor por una contraseña segura de su elección.

1. CREATE USER 'example\_user'@'%' IDENTIFIED WITH mysql\_native\_password BY 'password';

Ahora, debemos darle permiso a este usuario a la base de datos example\_database:

1. GRANT ALL ON example\_database.\* TO 'example\_user'@'%';

Esto proporcionará al usuario **example\_user** privilegios completos sobre la base de datos **example\_database** y, al mismo tiempo, evitará que este usuario cree o modifique otras bases de datos en su servidor.

Ahora, cierre el shell de MySQL con lo siguiente:

1. exit

Puede verificar si el usuario nuevo tiene los permisos adecuados al volver a iniciar sesión en la consola de MySQL, esta vez, con las credenciales de usuario personalizadas:

1. mysql -u example\_user -p

Observe el indicador -p en este comando, que le solicitará la contraseña que utilizó cuando creó el usuario **example\_user**. Después de iniciar sesión en la consola de MySQL, confirme que tenga acceso a la base de datos **example\_database**:

1. SHOW DATABASES;

Con esto se generará el siguiente resultado:

Output

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

| Database |

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

| example\_database |

| information\_schema |

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

2 rows in set (0.000 sec)

A continuación, crearemos una tabla de prueba denominada **todo\_list**: Desde la consola de MySQL, ejecute la siguiente instrucción:

1. CREATE TABLE example\_database.todo\_list (
2. item\_id INT AUTO\_INCREMENT,
3. content VARCHAR(255),
4. PRIMARY KEY(item\_id)
5. );

Inserte algunas filas de contenido en la tabla de prueba. Es posible que quiera repetir el siguiente comando algunas veces, usando valores diferentes:

1. INSERT INTO example\_database.todo\_list (content) VALUES ("My first important item");

Para confirmar que los datos se guardaron correctamente en su tabla, ejecute lo siguiente:

1. SELECT \* FROM example\_database.todo\_list;

Verá el siguiente resultado:

Output

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

| item\_id | content |

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

| 1 | My first important item |

| 2 | My second important item |

| 3 | My third important item |

| 4 | and this one more thing |

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

4 rows in set (0.000 sec)

Después de confirmar que haya datos válidos en su tabla de prueba, puede cerrar la consola de MySQL:

1. exit

Ahora, podrá crear una secuencia de comandos PHP que se conecte a MySQL y realice consultas relacionadas con su contenido. Cree un nuevo archivo PHP en su directorio web root personalizado usando su editor preferido. En este caso, usaremos nano:

1. nano /var/www/your\_domain/todo\_list.php

La siguiente secuencia de comandos PHP establece conexión con la base de datos de MySQL, realiza consultas relacionadas con el contenido de la tabla **todo\_list** y muestra los resultados en una lista. Si hay un problema con la conexión de la base de datos, generará una excepción. Copie este contenido en su secuencia de comandos todo\_list.php:

/var/www/your\_domain/todo\_list.php

**<?php**

$user = "example\_user";

$password = "password";

$database = "example\_database";

$table = "todo\_list";

try {

$db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);

echo "<h2>TODO</h2><ol>";

foreach($db->query("SELECT content FROM $table") as $row) {

echo "<li>" . $row['content'] . "</li>";

}

echo "</ol>";

} catch (PDOException $e) {

print "Error!: " . $e->getMessage() . "<br/>";

die();

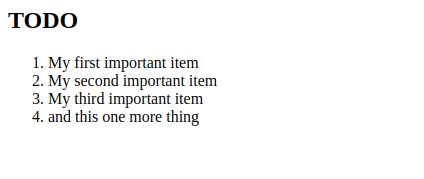
}

Guarde y cierre el archivo cuando finalice la edición.

Ahora, puede acceder a esta página en su navegador web al visitar el nombre de dominio o la dirección IP pública de su sitio web seguido de /todo\_list.php:

http://your\_domain/todo\_list.php

Debería ver una página como esta, en la que se muestra el contenido que insertó en su tabla de prueba:



Eso significa que su entorno PHP está listo para establecer conexión con su servidor de MySQL e interactuar con él.

**Conclusión**

En esta guía, creamos una base flexible para presentar sitios web y aplicaciones PHP a sus visitantes usando Apache como servidor web y MySQL como sistema de base de datos