

Instalamos NGINX, primero ponemos :sudo apt update.

```
root@david:/home/usuario# apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
25 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@david:/home/usuario#
```

Despues sudo apt install nginx

```
root@david:/home/usuario# apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 25 not upgraded.
Need to get 564 kB of archives.
After this operation, 1,596 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.5 [43.4 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
Fetched 564 kB in 1s (629 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 87294 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.5_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.5) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.5_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.5) ...
Setting up nginx-common (1.24.0-2ubuntu7.5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Setting up nginx (1.24.0-2ubuntu7.5) ...
 * Upgrading binary nginx
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@david:/home/usuario#
```

Para verificar qué perfiles UFW están disponibles sudo ufw app list

```
usuario@david:~$ sudo ufw app list
[sudo] password for usuario:
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
usuario@david:~$
```

Se recomienda que habilite el perfil más restrictivo, pero que aún permita el tráfico que necesita. Debido a que no configuró SSL para su servidor a través de esta guía, solo deberá permitir el tráfico de HTTP habitual en el puerto 80.

Puede habilitarlo escribiendo lo siguiente:

```
sudo ufw allow 'Nginx HTTP'
```

```
usuario@david:~$ sudo ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
usuario@david:~$
```

Puede verificar el cambio ejecutando lo siguiente:

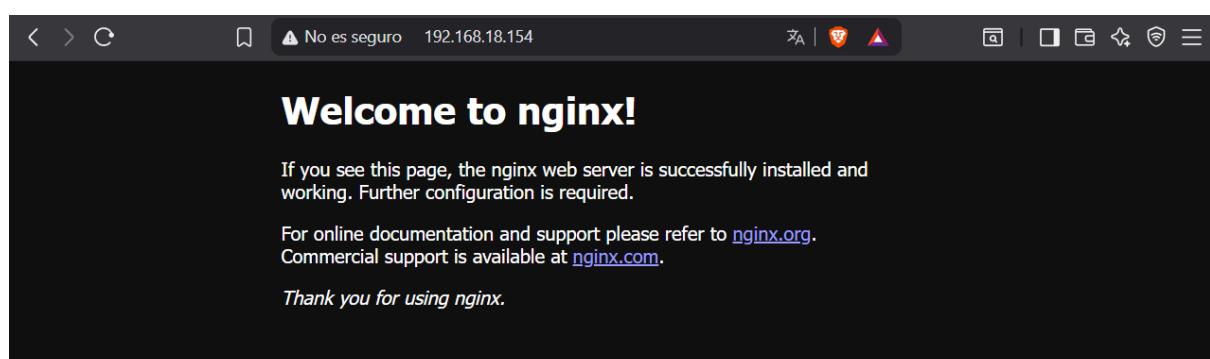
```
sudo ufw status
```

```
usuario@david:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
Nginx HTTP                 ALLOW       Anywhere
Nginx HTTP (v6)             ALLOW       Anywhere (v6)

usuario@david:~$
```

Ponemos nuestra IP en el navegador y debe aparecer esto.



# Instalar MySQL

Ponemos sudo apt install mysql-server

```
luisaricardo@david:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgifast-perl libcgipm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin
  libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite3t64 libtimedate-perl
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common
  mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl
  libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgifast-perl libcgipm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin
  libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite3t64 libtimedate-perl
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common
  mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 25 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1 [6,746 B]
Get:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [2,740 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [22.4 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libevent-pthreads-2.1-7t64 amd64 2.1.12-stable-9ubuntu2 [7,982 B]
Get:5 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libmecab2 amd64 0.996-14ubuntu4 [201 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 libprotobuf-lite3t64 amd64 3.21.12-8.2ubuntu0.2 [238 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server-core-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [2.17.5 MB]
Get:8 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [1,439 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libhtml-tagset-perl all 3.20-6 [11.3 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu noble/main amd64 liburi-perl all 5.27-1 [88.0 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libhtml-parser-perl amd64 3.81-1build3 [85.8 kB]
Get:12 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libcgipm-perl all 4.63-1 [185 kB]
Get:13 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 libfcgi0t64 amd64 2.4.2-2.1ubuntu0.24.04.1 [27.0 kB]
Get:14 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libfcgi-perl amd64 0.82+ds-3build2 [21.7 kB]
Get:15 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libfcgi-fast-perl all 1.2.17.1-1.2.17.1 [10.2 kB]
```

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: sudo mysql\_secure\_installation, le damos a yes y enter

```
luisaricardo@david:~$ sudo mysql_secure_installation
Securing the MySQL server deployment.
Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT 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 component?
Press y|Y for Yes, any other key for No: y
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
Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing and development and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

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) : y
Success.

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...

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.
```

Ponemos sudo mysql y debe de aparecer esto

```
usuario@david:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, 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>
```

Instalamos PHP

Para instalar los paquetes `php-fpm` y `php-mysql`, ejecute lo siguiente:

```
usuario@david:~$ sudo apt install php-fpm php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  php-common php8.3-cli php8.3-common php8.3-fpm php8.3-mysql php8.3-opcache php8.3-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  php-common php-fpm php-mysql php8.3-cli php8.3-common php8.3-fpm php8.3-mysql php8.3-opcache php8.3-readline
0 upgraded, 9 newly installed, 0 to remove and 25 not upgraded.
Need to get 5,112 kB of archives.
After this operation, 23.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Cree el directorio web root para `your_domain` de la siguiente manera:

`sudo mkdir /var/www/your_domain` y `sudo nano`

`/etc/nginx/sites-available/your_domain`

```
usuario@david:~$ sudo mkdir /var/www/arredondo
usuario@david:~$ sudo nano /etc/nginx/sites-available/arredondo
usuario@david:~$ usuario@David:~$
```

Modificamos el archivo

```
GNU nano 7.2                               /etc/nginx/sites-available/arredondo
server {
    listen 80;
    server_name arredondo www.arredondo;
    root /var/www/arredondo;

    index index.html index.htm index.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Creamos el enlace simbólico : sudo ln -s /etc/nginx/sites-available/your\_domain /etc/nginx/sites-enabled/

```
usuario@david:~$ sudo ln -s /etc/nginx/sites-available/arredondo /etc/nginx/sites-enabled/
```

Comprobamos que todo esté bien sintácticamente : sudo nginx -t

```
usuario@david:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
usuario@david:~$ client loop: send disconnect: Connection reset
```

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 nuevo bloque del servidor funcione según lo previsto: sudo nano /var/www/your\_domain/index.html

```
usuario@david:~$ usuario@david:~$ sudo nano /var/www/arredondo/index.html
```

Incluya el siguiente contenido en este archivo:

```
GNU nano 7.2                               /var/www/arredondo/index.html *
<html>
  <head>
    <title>arredondo</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is the landing page of <strong>arredondo</strong>.</p>
  </body>
</html>
```

Ponemos nuestro sitio en el navegador y debe de aparecer el index.html que hemos creado anteriormente.



# Probar PHP con Nginx

En el editor de texto, abra un archivo nuevo denominado `info.php` en el root de su documento:`sudo nano /var/www/your_domain/info.php`

```
usuario@david:~$ sudo nano /var/www/arredondo/info.php
```

En ella se incluye código PHP válido que mostrará información sobre su servidor:

```
GNU nano 7.2                               /var/www/arredondo/info.php *
<?php
phpinfo();-
```

Ponemos esto en el navegador y tendrá que aparecer lo siguiente

`http://server_domain_or_IP/info.php`

The screenshot shows a web browser window with the URL `192.168.0.205/info.php`. The title bar says "PHP Version 8.3.6". The page content is a table of PHP configuration information:

System	Linux david 6.8.0-86-generic #87-Ubuntu SMP PREEMPT_DYNAMIC Mon Sep 22 18:00:00 UTC 2025
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/fpm
Loaded Configuration File	/etc/php/8.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/fpm/conf.d
Additional .ini files parsed	/etc/php/8.3/fpm/conf.d/10-mysqlnd.ini, /etc/php/8.3/fpm/conf.d/10-opcache.ini, /etc/php/8.3/fpm/conf.d/20-calendar.ini, /etc/php/8.3/fpm/conf.d/20-ctype.ini, /etc/php/8.3/fpm/conf.d/20-fil.ini, /etc/php/8.3/fpm/conf.d/20-finfo.ini, /etc/php/8.3/fpm/conf.d/20-gettext.ini, /etc/php/8.3/fpm/conf.d/20-iconv.ini, /etc/php/8.3/fpm/conf.d/20-pdo_mysql.ini, /etc/php/8.3/fpm/conf.d/20-phar.ini, /etc/php/8.3/fpm/conf.d/20-readline.ini, /etc/php/8.3/fpm/conf.d/20-shmop.ini, /etc/php/8.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.3/fpm/conf.d/20-sysvsem.ini, /etc/php/8.3/fpm/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no

Borramos el archivo, por que tenemos información propia `sudo rm`

`/var/www/your_domain/info.php`

```
usuario@david:~$ sudo rm /var/www/arredondo/info.php
```

## Probar la conexión con la base de datos desde PHP

Primero, establezca conexión con la consola de MySQL usando la cuenta root:  
sudo mysql y despues CREATE DATABASE example\_database;

```
usuario@david:~$ sudo mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, 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> CREATE DATABASE arredondo;
```

Creamos el usuario y la contraseña

```
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'david'@'%' IDENTIFIED WITH mysql_native_password BY 'asdf,123';
```

Ahora, debemos darle permiso a este usuario a la base de datos `example_database`:

```
mysql> GRANT ALL ON arredondo.* TO 'david'@'%';
```

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: mysql -u example\_user -p

```
usuario@david:~$ mysql -u david -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, 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.
```

Con esto se generará el siguiente resultado: SHOW DATABASES;

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| arredondo |
| information_schema |
| performance_schema |
+-----+
3 rows in set (0.01 sec)

mysql>
```

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

```
mysql> CREATE TABLE arredondo.todo_list (
    -> item_id INT AUTO_INCREMENT,
    -> content VARCHAR(255),
    -> PRIMARY KEY(item_id)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

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

```
mysql> INSERT INTO arredondo.todo_list (content) VALUES ("My first important item");
Query OK, 1 row affected (0.01 sec)
```

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

```
mysql> SELECT * FROM arredondo.todo_list;
+-----+
| item_id | content          |
+-----+
|      1  | My first important item |
+-----+
1 row in set (0.00 sec)
```

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

```
by
usuario@david:~$ sudo nano /var/www/arredondo/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:

```
GNU nano 7.2                               /var/www/arredondo/todo_list.php
<?php
$user = "david";
$password = "Asdf,123";
$database = "arredondo";
$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();
}
```

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



## TODO

1. My first important item