

Installation d'un serveur (LAMP) de type Apache

Un serveur LAMP est un environnement de développement et d'hébergement web populaire composé de quatre composants principaux :

1. Linux : le système d'exploitation qui sert de base au serveur
2. Apache : le serveur web qui gère les requêtes http
3. MariaDB (ou MySQL) : le système de gestion de base de données relationnelle qui Stocke les données du site web.
4. PHP : le langage de programmation utilisé pour générer dynamiquement le contenu des pages web

Lamp est open-source, il est souvent utilisé pour héberger des applications web tel que des sites e-commerce ou des systèmes de gestion de contenu.

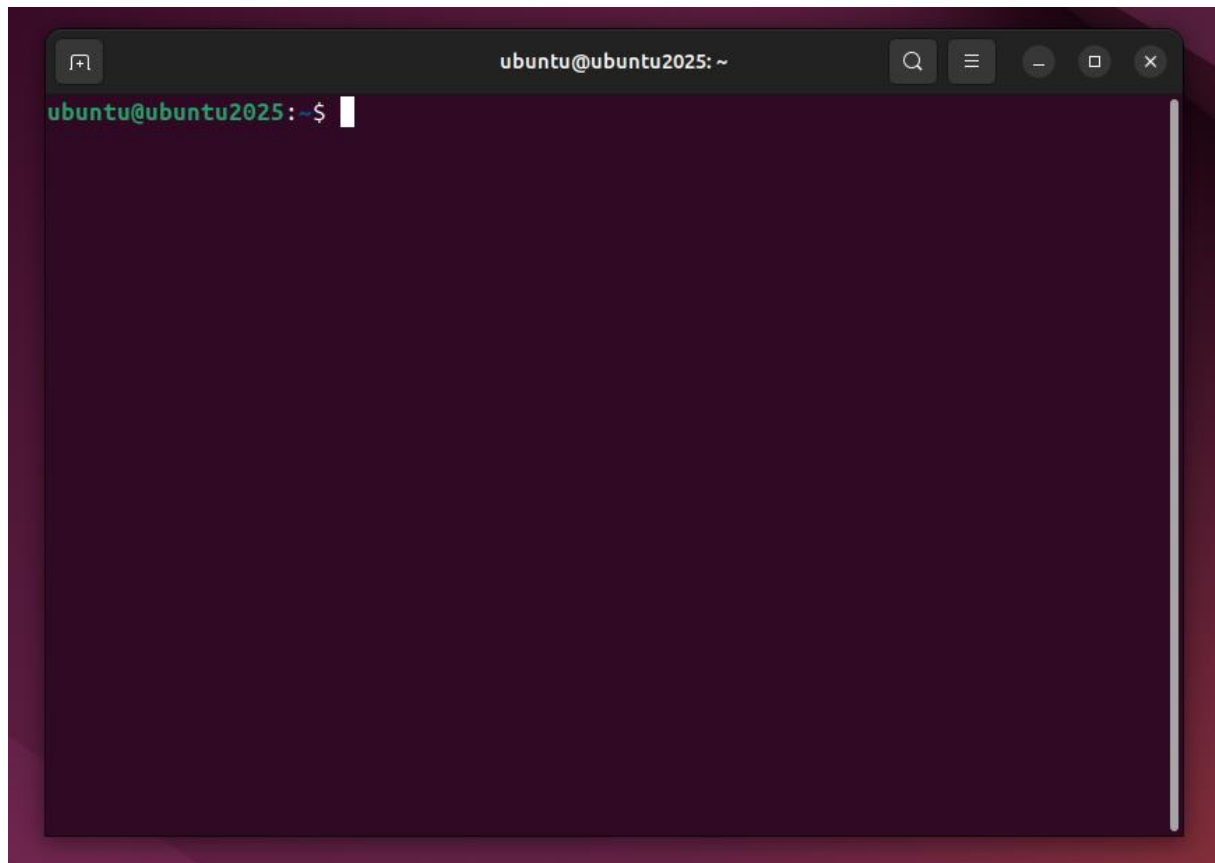
Nous allons maintenant voir comment installer et configurer un environnement LAMP complet :

Prérequis :

- Avoir le logiciel de virtualisation VirtualBox
- Avoir installer un fichier ISO Ubuntu

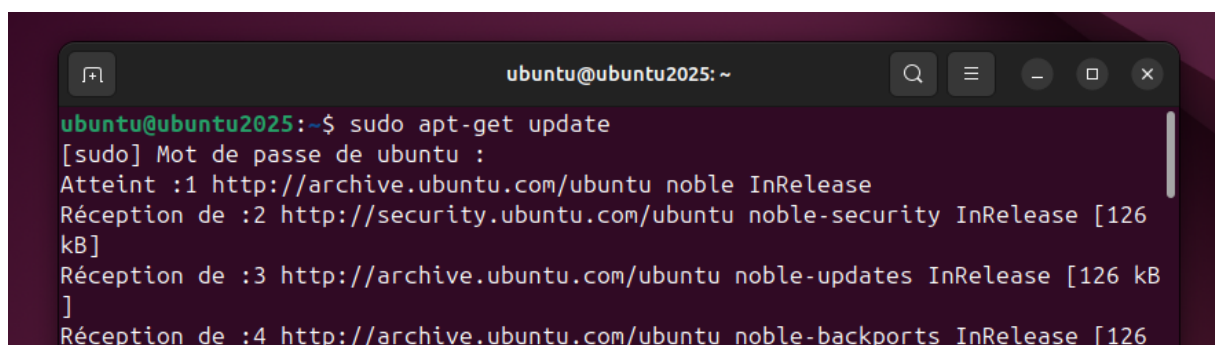
Etape 1 :

- Ouvrez le terminal en appuyant simultanément sur les touches **ctrl+alt+t** :

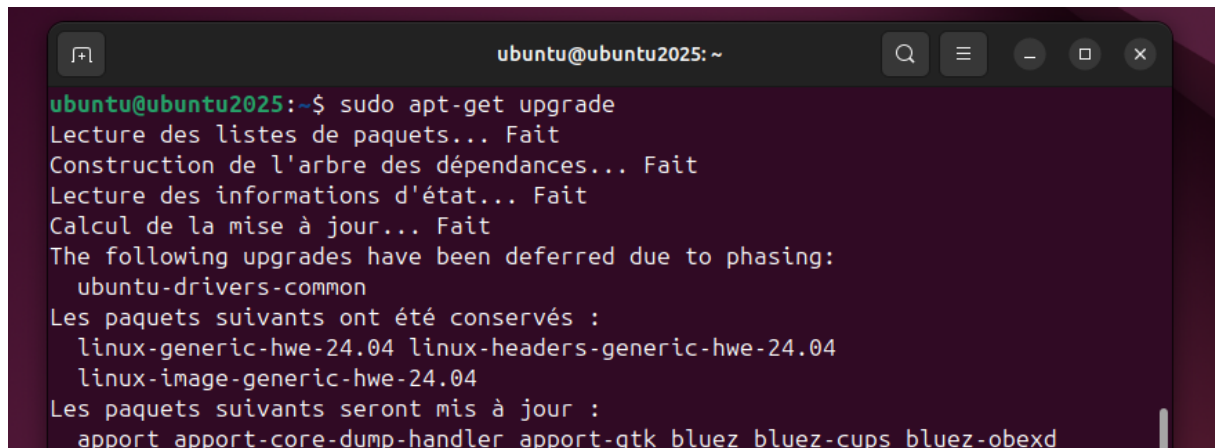


- Nous allons mettre à jour des référentiels à l'aide la commande suivante :

sudo apt-get update



- Ensuite, tapez la commande : **sudo apt-get upgrade** pour installer les améliorations de tous les packages.

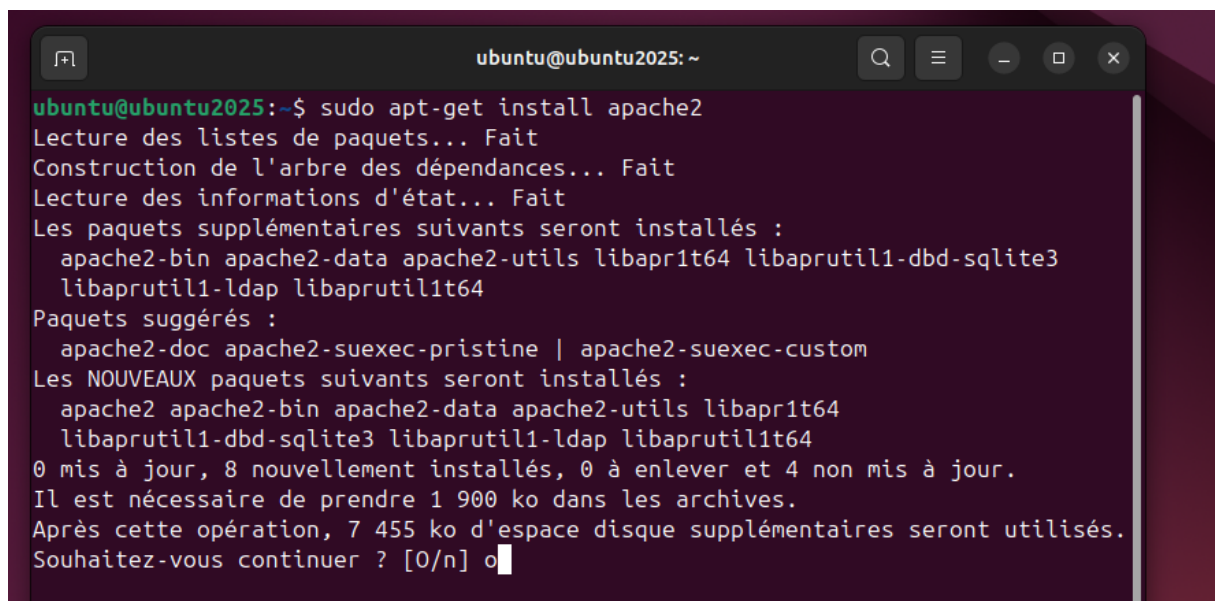
A terminal window titled 'ubuntu@ubuntu2025: ~' showing the output of the command 'sudo apt-get upgrade'. The output indicates that several upgrades have been deferred due to phasing, specifically 'ubuntu-drivers-common'. It also lists packages that have been preserved ('linux-generic-hwe-24.04', 'linux-headers-generic-hwe-24.04', 'linux-image-generic-hwe-24.04') and packages that will be updated ('appport', 'appport-core-dump-handler', 'appport-gtk', 'bluez', 'bluez-cups', 'bluez-obexd').

```
ubuntu@ubuntu2025:~$ sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
The following upgrades have been deferred due to phasing:
  ubuntu-drivers-common
Les paquets suivants ont été conservés :
  linux-generic-hwe-24.04 linux-headers-generic-hwe-24.04
  linux-image-generic-hwe-24.04
Les paquets suivants seront mis à jour :
  appport appport-core-dump-handler appport-gtk bluez bluez-cups bluez-obexd
```

Etape 2:

La prochaine étape consiste à installer le serveur web Apache

- Tapez la commande **sudo apt-get install apache2** et acceptez le téléchargement avec la touche **o** pour oui :

A terminal window titled 'ubuntu@ubuntu2025: ~' showing the output of the command 'sudo apt-get install apache2'. The output lists additional packages to be installed along with apache2, such as 'apache2-bin', 'apache2-data', 'apache2-utils', and various 'libapr' and 'libaprutil' packages. It also shows suggested packages and the disk space requirements for the installation.

```
ubuntu@ubuntu2025:~$ sudo apt-get install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-bin apache2-data apache2-utils libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
0 mis à jour, 8 nouvellement installés, 0 à enlever et 4 non mis à jour.
Il est nécessaire de prendre 1 900 ko dans les archives.
Après cette opération, 7 455 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] o
```

- Vérifiez ensuite le statuts du serveur afin de s'assurer de son fonctionnement via la commande **systemctl status apache2** :

```
ubuntu@ubuntu2025:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
   Active: active (running) since Tue 2025-03-25 17:26:06 UTC; 14min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 18787 (apache2)
      Tasks: 55 (limit: 4610)
     Memory: 5.6M (peak: 5.9M)
        CPU: 128ms
    CGroup: /system.slice/apache2.service
            └─18787 /usr/sbin/apache2 -k start
              └─18788 /usr/sbin/apache2 -k start
                └─18790 /usr/sbin/apache2 -k start

mars 25 17:26:06 ubuntu2025 systemd[1]: Starting apache2.service - The Apache H>
```

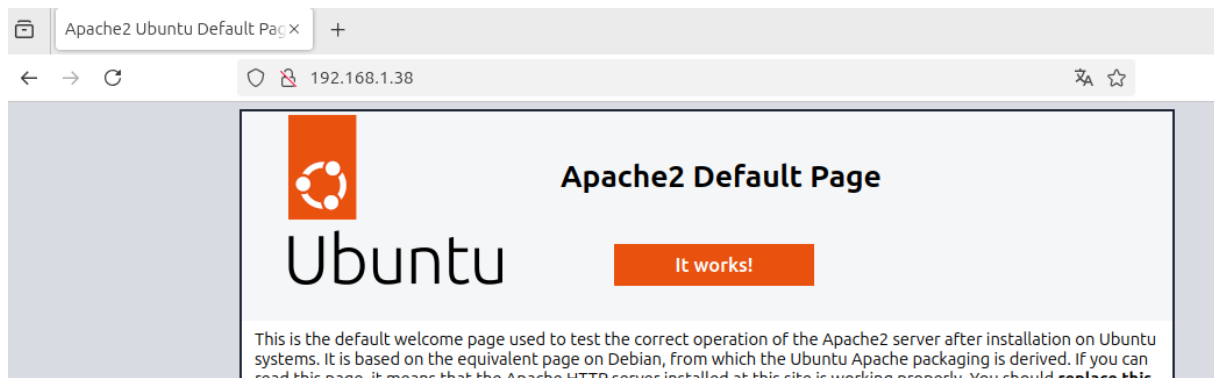
Le serveur fonctionne correctement, il faut aussi vérifier si il fonctionne bien depuis notre navigateur .

- Allez récupérer votre adresse IP en tapant la commande **ip a**

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:01:6e:53 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.38/24 brd 192.168.1.255 scope global dynamic noprefixroute en
```

Nous voyons que le nom de notre interface réseau se nomme enp0s3 et que l'adresse IP de notre machine est : 192.168.1.38

- Après avoir vérifié l'adresse IP, nous allons la rentrer dans notre navigateur :



Etape 3:

Nous allons maintenant installer les paquets du module php :

- Revenez sur votre terminal et tapez : **sudo apt-get install php**

```
ubuntu@ubuntu2025: ~  
ubuntu@ubuntu2025:~$ sudo apt-get install php  
[sudo] Mot de passe de ubuntu :  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  libapache2-mod-php8.3 libsodium23 php-common php8.3 php8.3-cli php8.3-common  
  php8.3-opcache php8.3-readline  
Paquets suggérés :  
  php-pear  
Les NOUVEAUX paquets suivants seront installés :  
  libapache2-mod-php8.3 libsodium23 php php-common php8.3 php8.3-cli  
  php8.3-common php8.3-opcache php8.3-readline  
0 mis à jour, 9 nouvellement installés, 0 à enlever et 4 non mis à jour.  
Il est nécessaire de prendre 5 076 ko dans les archives.  
Après cette opération, 22,8 Mo d'espace disque supplémentaires seront utilisés.  
Souhaitez-vous continuer ? [0/n] o
```

- Installez ensuite l'extension cURL avec la commande **sudo apt-get install curl**

L'extension curl est comme une sorte de bibliothèque qui facilite la communication entre les différents types de serveurs dit applicatifs

```
ubuntu@ubuntu2025: ~  
ubuntu@ubuntu2025:~$ sudo apt-get install curl  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les NOUVEAUX paquets suivants seront installés :  
  curl  
0 mis à jour, 1 nouvellement installés, 0 à enlever et 4 non mis à jour.  
Il est nécessaire de prendre 226 ko dans les archives.  
Après cette opération, 534 ko d'espace disque supplémentaires seront utilisés.  
Réception de :1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 curl a  
md64 8.5.0-2ubuntu10.6 [226 kB]  
226 ko réceptionnés en 6s (39,5 ko/s)  
Sélection du paquet curl précédemment désélectionné.  
(Lecture de la base de données... 151900 fichiers et répertoires déjà installés.  
)  
Préparation du dépaquetage de .../curl_8.5.0-2ubuntu10.6_amd64.deb ...  
Dépaquetage de curl (8.5.0-2ubuntu10.6) ...  
Paramétrage de curl (8.5.0-2ubuntu10.6) ...  
Traitement des actions différées (« triggers ») pour man-db (2.12.0-4build2) ...  
ubuntu@ubuntu2025:~$
```

Etape 4 :

La suite logique de l'installation d'un serveur LAMP est l'installation d'un système de gestion de base de donnée, ce système se nomme mariaDB.

- Tapez la commande **sudo apt-get install mariadb-server** puis **o** pour continuer l'installation :

```
ubuntu@ubuntu2025: ~  
ubuntu@ubuntu2025:~$ sudo apt-get install mariadb-server  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  galera-4 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl  
  libdbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl libfcgi0t64  
  libhtml-template-perl libmariadb3 libmysqlclient21 libsigsegv2 libsnappy1v5  
  libterm-readkey-perl liburing2 mariadb-client mariadb-client-core
```

Toujours dans la continuité, nous allons installer le serveur MySQL avec la commande :
sudo mysql_secure_installation

```
ubuntu@ubuntu2025:~$ mysql_secure_installation
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
haven't set the root password yet, you should just press enter here.
```

```
Enter current password for root (enter for none):
```

Ne mettez pas de mot de passe pour le moment car il n'a pas encore été créé, appuyer sur entrée

- Nous allons ajouter un mot de passe en appuyant sur y, et sur les étapes suivantes appuyer sur y également.

```
Disallow root login remotely? [Y/n] y  
... Success!
```

```
By default, MariaDB 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? [Y/n] 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? [Y/n] y  
... Success!
```

```
Cleaning up...
```

```
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.
```

L'environnement complet est à présent installé !

Configuration de mariaDB :

La plupart des projets logiciels et Web nécessitent l'utilisation d'une base de données où les informations peuvent être stockées, classées et conservées prêtes à être récupérées à tout moment par le biais de consultations. MySQL est sans aucun doute la base de données la plus utilisée pour les projets Web et autres types de programmes, mais il existe des alternatives telles que MariaDB qui sont très puissantes, flexibles et basées sur des modèles de logiciels libres.

Etape 1 :

Pour commencer : nous allons ouvrir une session dans le terminal à l'aide de la commande **mysql -u root -p** puis entrez votre mot de passe :

```
ubuntu@ubuntu2025:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.11.7-MariaDB-2ubuntu2 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> 
```

Nous allons tester le serveur en créant une base de donnée, je l'appellerai TESTDATA:

- Entrez la commande **create database TESTDATA ;**

```
MariaDB [(none)]> create database TESTDATA ;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> 
```

- Nous allons nous y connecter en tapant use **TESTDATA ;**

```
MariaDB [(none)]> use TESTDATA ;
Database changed
MariaDB [TESTDATA]> 
```


Etape 2 :

Dans cette nouvelle base de donnée, nous allons créer une table nommé test avec la commande **create table test (msg text) ;**

```
MariaDB [(none)]> use TESTDATA ;  
Database changed  
MariaDB [TESTDATA]> create table test (msg text) ;  
Query OK, 0 rows affected (0,026 sec)  
  
MariaDB [TESTDATA]>
```

- nous allons donc insérer une donnée grâce à la commande :

insert into test values ('salut') ;

```
MariaDB [TESTDATA]> insert into test values ('salut') ;  
Query OK, 1 row affected (0,015 sec)  
  
MariaDB [TESTDATA]> █
```

- Effectuons un maintenant un test pour contrôler si le message s'affiche :

Select * from test ;

```
MariaDB [TESTDATA]> select * from test ;  
+-----+  
| msg    |  
+-----+  
| salut  |  
+-----+  
1 row in set (0,002 sec)  
  
MariaDB [TESTDATA]> █
```

Nous pouvons quitter le serveur grâce à la commande **quit ;**

```
MariaDB [TESTDATA]> quit ;  
Bye  
ubuntu@ubuntu2025:~$
```

Création de fichier PHP :

nous allons créer un fichier php qui fera appel à la fonction test.php.

- déplaçons nous dans le répertoire /var/www/html grâce à la commande cd :

cd /var/www/html

```
ubuntu@ubuntu2025:~$ cd /var/www/html
ubuntu@ubuntu2025:/var/www/html$
```

Tapez ensuite **ls** pour lister les fichiers contenus dans le dossier :

```
ubuntu@ubuntu2025:/var/www/html$ ls
index.html
```

Et taper touch test.php pour créer un fichier PHP :

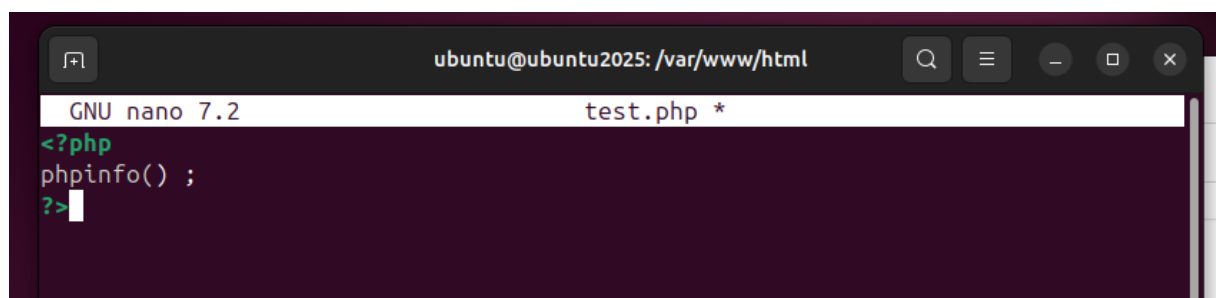
```
ubuntu@ubuntu2025:/var/www/html$ sudo touch test.php
[sudo] Mot de passe de ubuntu :
ubuntu@ubuntu2025:/var/www/html$ ls
index.html  test.php
ubuntu@ubuntu2025:/var/www/html$
```

- Et modifiez-le en tapant : **sudo nano test.php**
- Entrez les lignes suivantes dans le fichier :

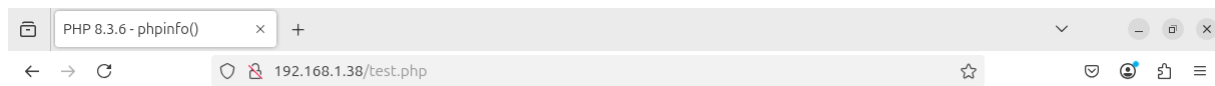
<?php

Phpinfo() ;

?>



- Sauvegardez en faisant **ctrl + x** pour sauver l'espace modifier, puis **o** pour oui et enfin **entrée** pour quitter le fichier :
- Enfin aller sur votre navigateur et entrez l'adresse IP que vous êtes allé chercher précédemment en ajoutant /test.php



PHP Version 8.3.6	
System	Linux ubuntu2025 6.11.0-17-generic #17~24.04.2-Ubuntu SMP PREEMPT_DYNAMIC Mon Jan 20 22:48:29 UTC 2 x86_64
Build Date	Dec 2 2024 12:36:18
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	40230831
Zend Extension Build	API40230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Non seulement le fichier à été trouvé mais il fonctionne parfaitement.