

Manuel d'installation : LAMP

Avant de commencer le didacticiel Ubuntu LAMP, vous aurez besoin :

- D'une machine locale avec le client **SSH** installé
- D'un **VPS faisant tourner Ubuntu 16.04**
- D'un utilisateur non root avec les privilèges **sudo**

Il est recommandé d'utiliser un utilisateur **sudo** au lieu d'un utilisateur **root** pour installer un logiciel pour des raisons de sécurité. Si vous avez un accès root à votre serveur VPS, vous pouvez créer un utilisateur sudo avec les commandes suivantes :

```
adduser <username>
```

La commande ci-dessus crée un utilisateur avec votre nom d'utilisateur fourni. Maintenant, faites de ce nouvel utilisateur créé un utilisateur sudo

```
usermod -aG sudo <username>
```

Passez sur ce nouvel utilisateur.

```
sudo su - <username>
```

Étape 1 – Installation du serveur Web Apache

Avant de commencer l'installation, mettez à jour votre système et assurez-vous d'avoir les derniers paquets.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Installez Apache2 avec la commande suivante :

```
sudo apt-get install apache2
```

Vérification de votre installation

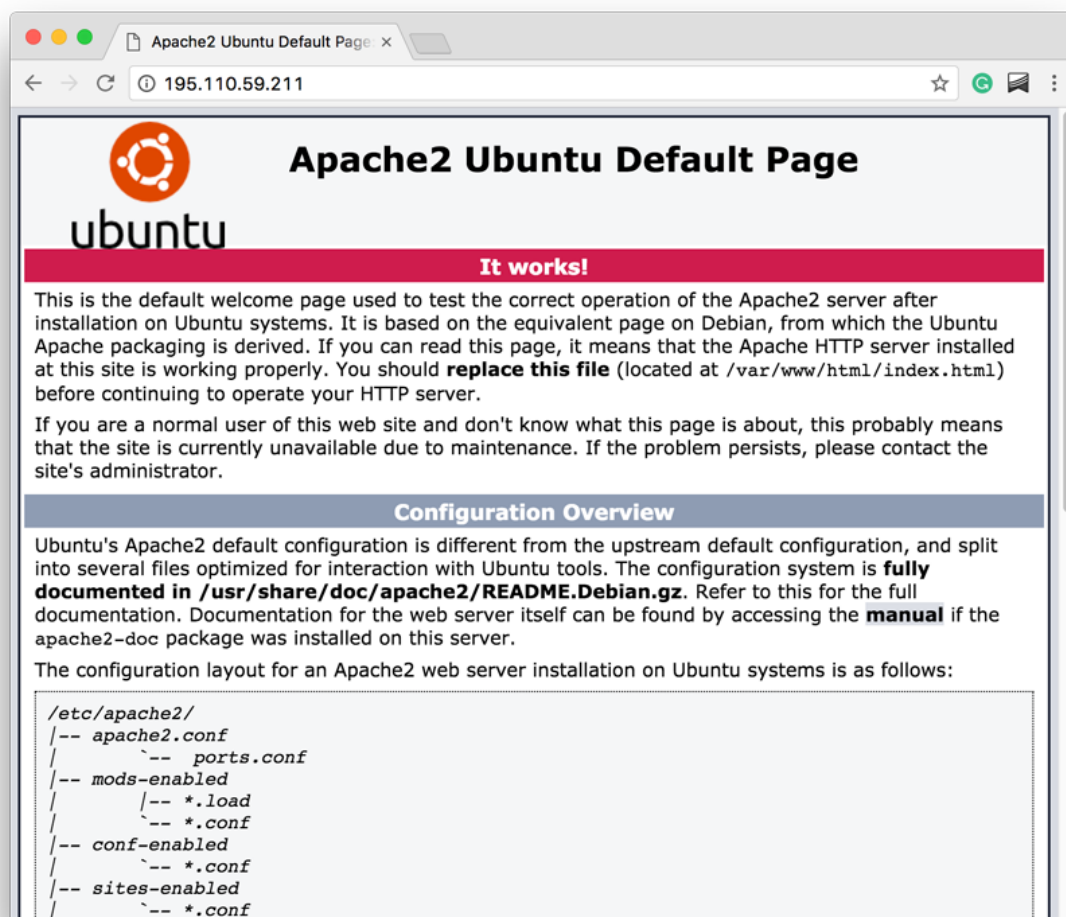
Pour vérifier l'installation, ouvrez votre navigateur sur votre machine locale et entrez l'adresse suivante dans la barre d'adresse.

http://<your_vps_ip_address_here>

Par exemple, si votre adresse IP VPS est 22.23.24.45, votre adresse doit être :

<http://195.110.59.211>

Vous devriez voir une page qui ressemble à ceci :



Remarque : si vous ne connaissez pas l'adresse IP de votre VPS, le moyen le plus rapide de le trouver est de saisir la commande suivante. **Cette commande imprime l'adresse IP publique de votre VPS.**

```
dig +short myip.opendns.com @resolver1.opendns.com
```

Dépannage de votre installation

Si vous n'avez pas vu l'image ci-dessus, ne vous inquiétez pas, vous avez peut-être activé le **pare-feu**. Vous devez activer Apache pour diffuser des requêtes Web sur le port 80 et le port 443 dans votre pare-feu. Installez UFW.

```
sudo apt-get install ufw
```

Ensuite, autorisez le trafic HTTP et HTTPS à travers le pare-feu.

```
sudo ufw allow http
```

```
sudo ufw allow https
```

Cette commande autorise le trafic HTTP et HTTPS via le pare-feu. **UFW** est une application de ligne de commande appelée Uncomplicated Firewall. Elle est utilisée pour gérer et créer des règles pour le pare-feu Linux. Entrez maintenant votre adresse IP VPS dans votre navigateur pour vérifier l'installation. Vous pouvez vérifier le statut du serveur Apache avec la commande suivante.

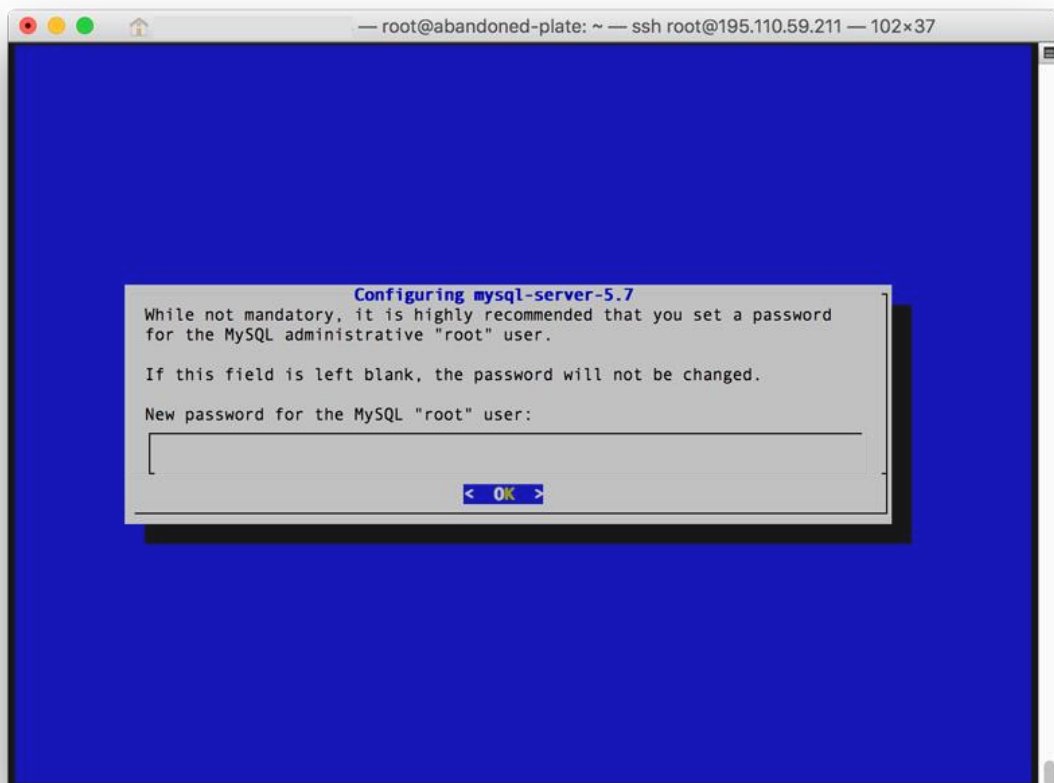
```
sudo systemctl status apache2
```

Étape 2 – Installation de MySQL

MySQL est la base de données de votre application. Pour installer MySQL, entrez la commande suivante.

```
sudo apt-get install mysql-server
```

Pendant l'installation, il vous demandera le mot de passe de l'utilisateur root. Assurez-vous d'utiliser un mot de passe fort. **Ne le laissez pas vide.**



L'utilisateur **root** est l'utilisateur privilégié le plus élevé de MySQL. En utilisant l'utilisateur root, vous pouvez créer d'autres utilisateurs pour les bases de données. Vous pouvez vérifier l'état du service MySQL avec la commande suivante :

```
sudo systemctl status mysql
```

Exemple de résultat :

```
mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor
  preset: enabled)
  Active: active (running) since Tue 2017-07-11 09:39:40 EDT; 1min 39s ago
  Main PID: 9579 (mysqld)
  CGroup: /system.slice/mysql.service
          └─9579 /usr/sbin/mysqld

Jul 11 09:39:39 abandoned-plate systemd[1]: Starting MySQL Community
  Server...
Jul 11 09:39:40 abandoned-plate systemd[1]: Started MySQL Community Server.
```

Étape 3 – Installation de PHP

PHP exécute votre application. Installez PHP et les modules supplémentaires avec la commande suivante

```
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql php-cgi php-curl php-json
```

Cette commande installera la dernière version stable de PHP et certains modules supplémentaires requis pour que l'application Web fonctionne.

Étape 3.1 – Vérification de votre installation PHP

Maintenant que vous avez installé PHP, nous allons vérifier s'il fonctionne correctement en créant un fichier de test et en l'ouvrant dans le navigateur. D'abord, installez l'éditeur de texte **nano**.

```
sudo apt-get install nano
```

Nano est un éditeur de texte de ligne de commande et est plus facile à démarrer pour les débutants. Maintenant, entrez la commande suivante.

```
sudo nano /var/www/html/test.php
```

Cette commande ouvrira l'éditeur nano avec un fichier **test.php** vide pour l'édition. Le répertoire **/var/www/html** où nous créons notre fichier PHP test est connu sous le nom de **webroot**. C'est là que Apache recherche le fichier demandé dans l'URL du site Web par défaut s'il n'a pas été configuré pour regarder ailleurs. En outre, vous avez besoin

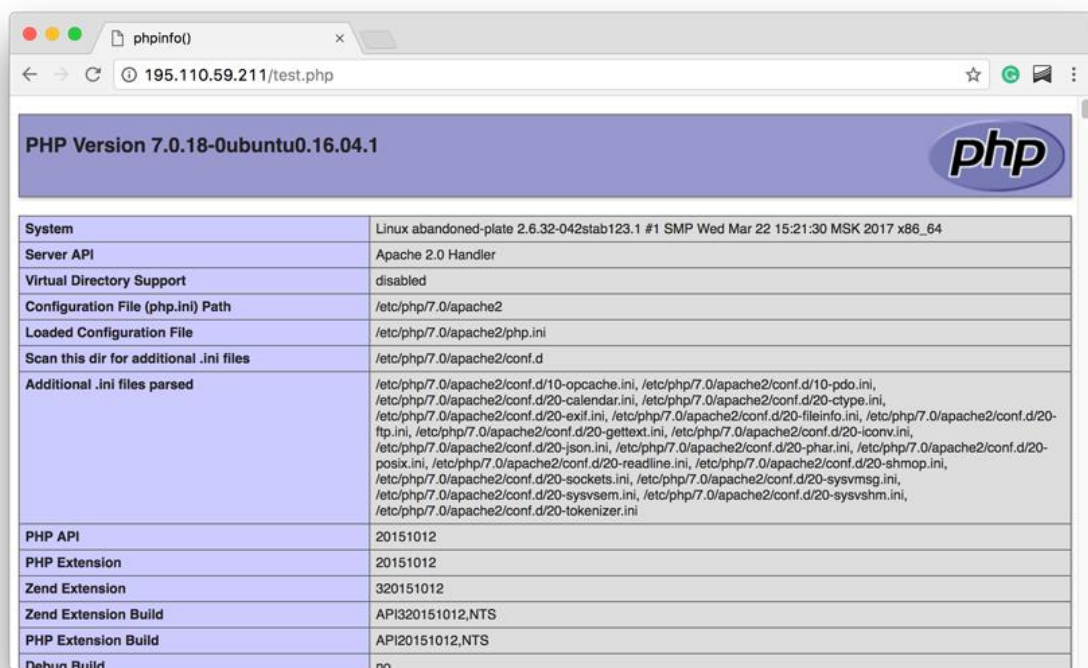
de privilèges **root** pour pouvoir écrire sur ce répertoire. Nous avons utilisé **sudo** avant notre commande. Entrez maintenant le texte suivant dans l'éditeur ouvert

```
<?php
phpinfo();
?>
```

Après avoir entré ce texte, appuyez sur **Ctrl + X** (ou **CMD + X** pour Mac), puis sur **Y**, puis **appuyez** sur **ENTRÉE**. Cela sauvegardera le fichier et quittera l'éditeur. Maintenant, ouvrez l'adresse Web suivante dans votre navigateur

`http://<your_vps_ip_adress>/test.php`

Vous devriez voir une page qui ressemble à ceci :



PHP Version 7.0.18-0ubuntu0.16.04.1	
System	Linux abandoned-plate 2.6.32-042stab123.1 #1 SMP Wed Mar 22 15:21:30 MSK 2017 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysmsg.ini, /etc/php/7.0/apache2/conf.d/20-syssem.ini, /etc/php/7.0/apache2/conf.d/20-sysshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no

La fonction **phpinfo ()** que nous avons appelée dans notre script **test.php** affiche des informations sur l'installation PHP et sa configuration. Maintenant, supprimez ce fichier de test en entrant la commande suivante:

```
sudo rm /var/www/html/test.php
```

Remarque : Il est très important de supprimer ce fichier de test après avoir vérifié l'installation, car il peut aider un hacker à obtenir des informations critiques sur la configuration du serveur.

Conclusion

Vous avez appris à installer LAMP sur Ubuntu. Après l'installation, vous pouvez copier vos fichiers PHP vers le serveur et déployer votre application Web. Vous pouvez également installer phpMyAdmin pour gérer vos bases de données sur une interface Web.