

Installer un serveur LAMP sous Debian Wheezy

Date de dernière mise à jour : 14/11/2014 à 17:29

Source : <http://www.vulgarisation-informatique.com/installer-serveur-lamp-debian.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Installer un serveur LAMP (Linux Apache MySQL PHP) n'est pas très compliqué. Nous allons voir comment installer un serveur web sous Debian Wheezy (7.x) avec PHP 5.5, MySQL 5.5 et PostgreSQL 9.3.2. Nous verrons comment installer sendmail pour envoyer des mails avec PHP, et configurerons aussi l'[URL Rewriting](#), ainsi que différents modules PHP qui pourront vous être utiles (PEAR notamment).

Ajout du dépôt dotdeb

Pour que Debian puisse savoir où se trouve le paquet de PHP 5.5 (vous pouvez aussi le faire avec PHP 5.4 en remplaçant 55 par 54 dans ce tutoriel), vous avez besoin de faire connaître les sources des paquets au gestionnaire de paquets.

Nous allons utiliser le dépôt de paquets Dotdeb, qui est un dépôt spécialisé pour serveur LAMP sous Debian.

Loggez-vous en tant que root avec la commande **su** - puis entrez le mot de passe root :

```
~  
#  
su -
```

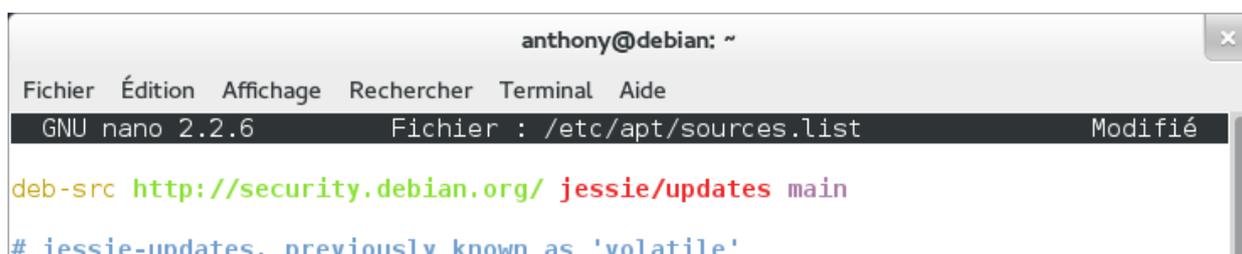
Pour des raisons de simplification, nous nous loggons en root dans un terminal. Peut-être n'avez-vous pas envie de procéder de la sorte. Faites dans ce cas précéder vos commandes par **sudo**.

Il va maintenant falloir modifier le fichier **sources.list** situé ici : **/etc/apt/sources.list**. Vous pouvez modifier ce fichier à l'aide de nano en tapant la commande suivante :

```
~  
#  
nano /  
etc/  
apt/  
sources.  
list  
Rajoutez les lignes suivantes :
```

```
deb http:  
//packages.dotdeb.org wheezy all  
deb-  
src http:  
//packages.dotdeb.org wheezy all  
deb http:  
//packages.dotdeb.org wheezy-php55 all  
deb-  
src http:  
//packages.dotdeb.org wheezy-php55 all
```

Ce qui donne ceci dans notre fichier :



```
anthony@debian: ~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
GNU nano 2.2.6      Fichier : /etc/apt/sources.list      Modifié  
deb-src http://security.debian.org/ jessie/updates main  
# jessie-updates, previously known as 'volatile'
```

Appuyez sur les touches **CTRL+X** puis sur **O** (oui) puis validez en appuyant sur **Entrée**. Le fichier a été sauvegardé.

Lorsque vous téléchargerez des paquets, vous aurez besoin que la clé du serveur distant soit approuvée (sinon, vous vous retrouverez avec une erreur **NO_PUBKEY / GPG error**). Pour ce faire, exécutez les commandes suivantes :

```
~
#
wget http:
//
www.dotdeb.org/dotdeb.gpg
~
#
cat dotdeb.
gpg |
apt-
key add -
```

Réalisez ensuite une mise à jour de la liste des paquets disponibles puis une mise à jour de ceux-ci en tapant les commandes suivantes :

```
~
#
apt-
get update~
#
apt-
get upgrade
```

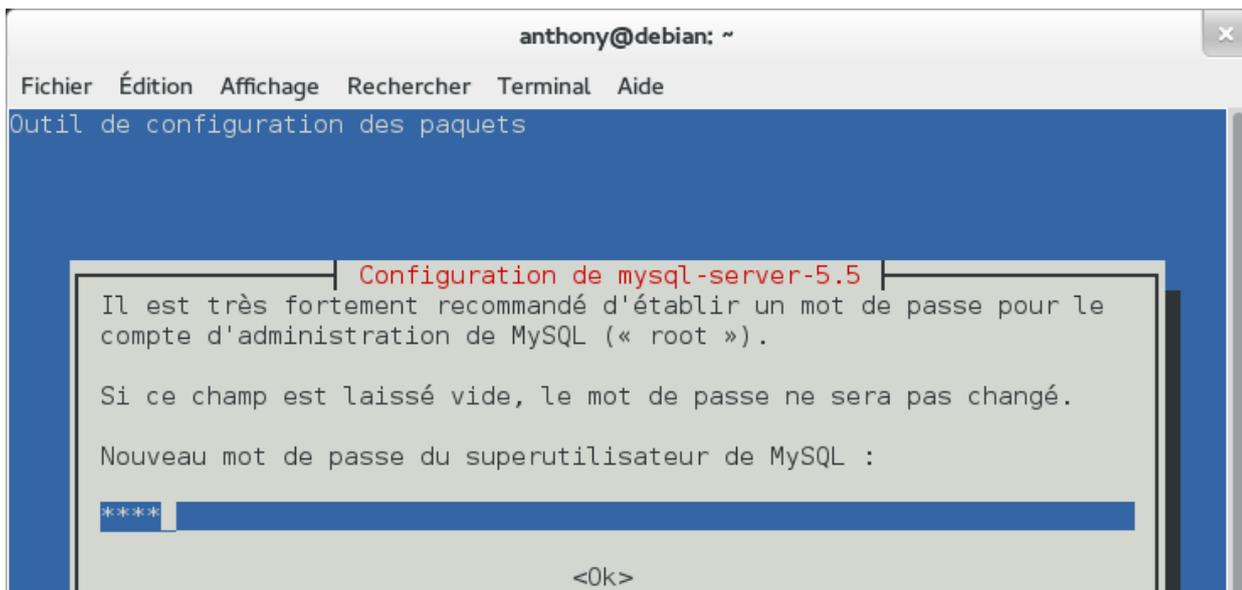
Répondez **Oui** (en appuyant sur la touche **O**) à chacune des questions qui peuvent vous être posées.

Installation du serveur LAMP

Nous allons installer Apache 2, PHP 5.5, MySQL 5.5 et PHPMyAdmin pour vous permettre de gérer vos bases de données MySQL. Pour ce faire, exécutez la commande suivante :

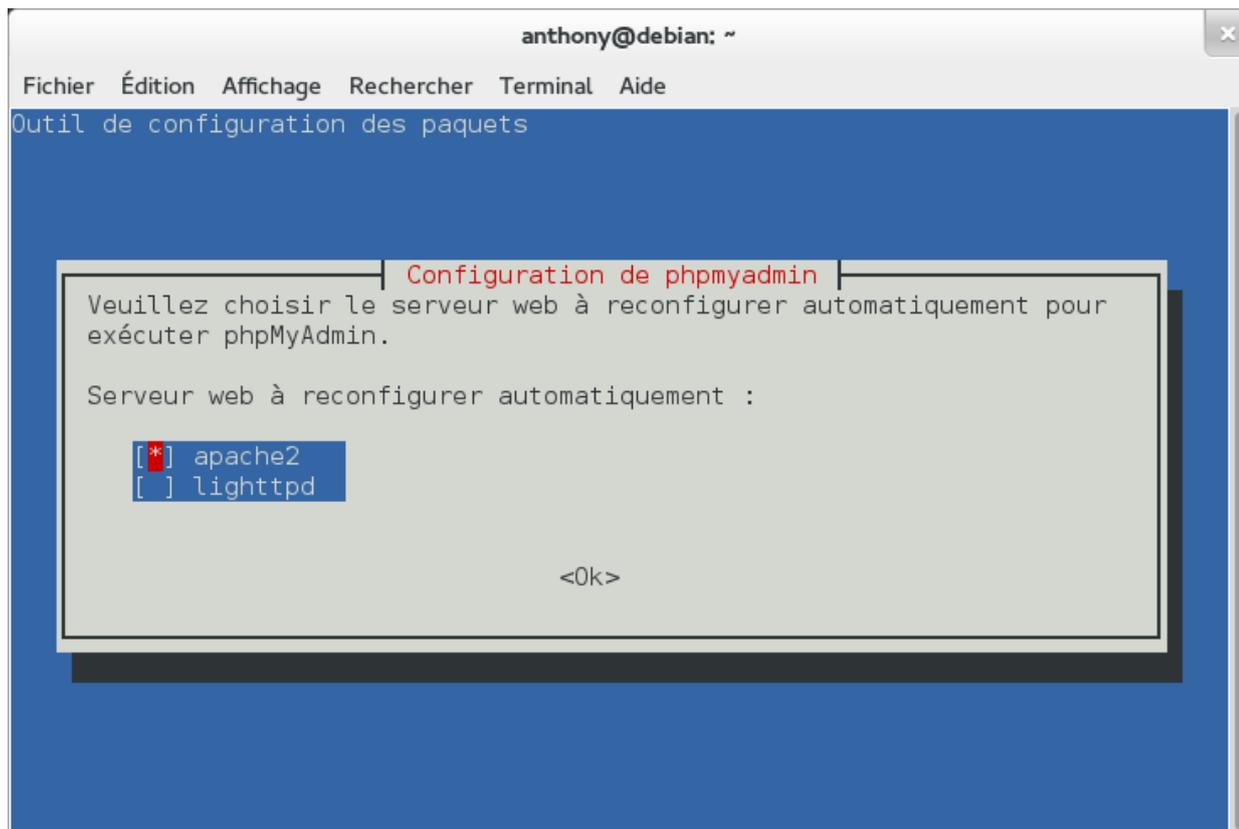
```
~
#
apt-
get install apache2 apache2-
mpm-
prefork php5 mysql-
server phpmyadmin
```

Répondez **Oui** à chacun des messages d'installation de paquets. Au cours de l'installation, vous allez devoir définir certains paramètres. La première fenêtre concerne le mot de passe root du serveur MySQL :

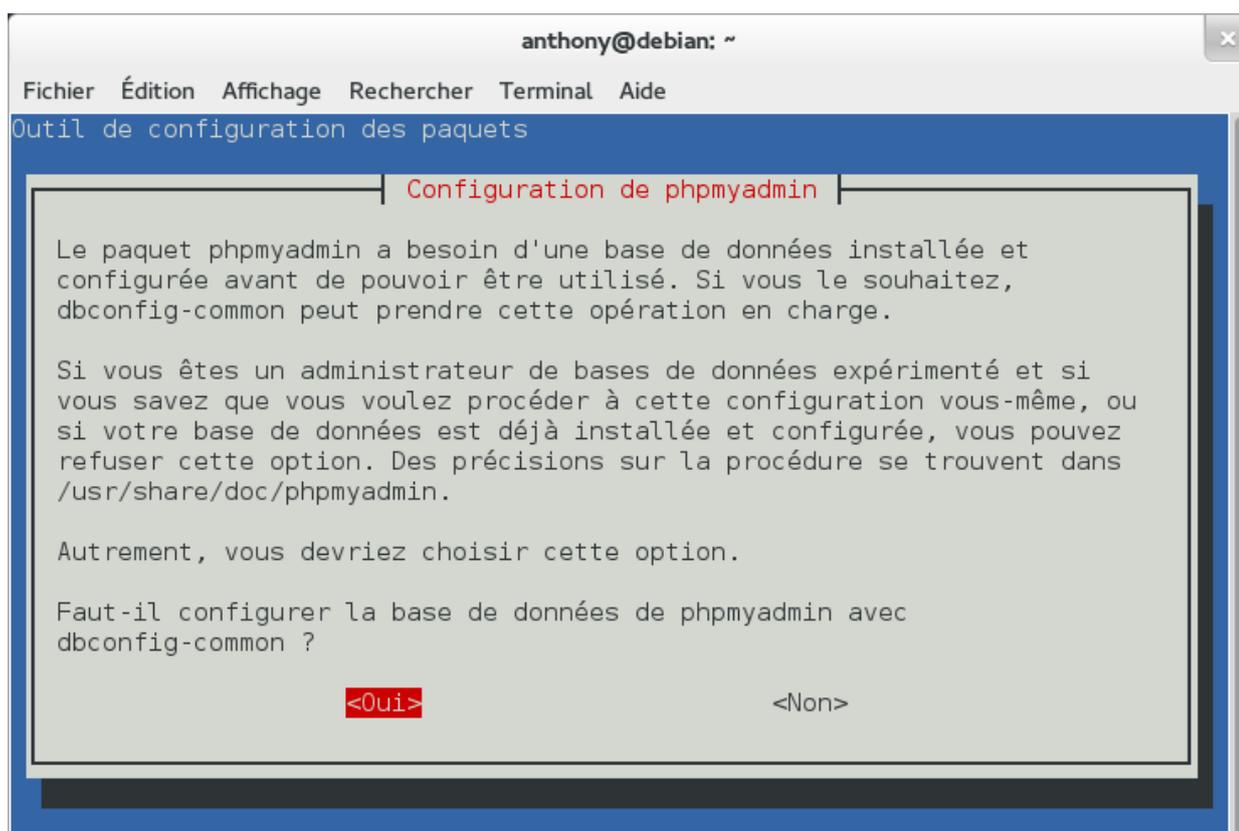


Choisissez un mot de passe pour votre serveur MySQL puis validez. Confirmez ce mot de passe puis validez une seconde fois.

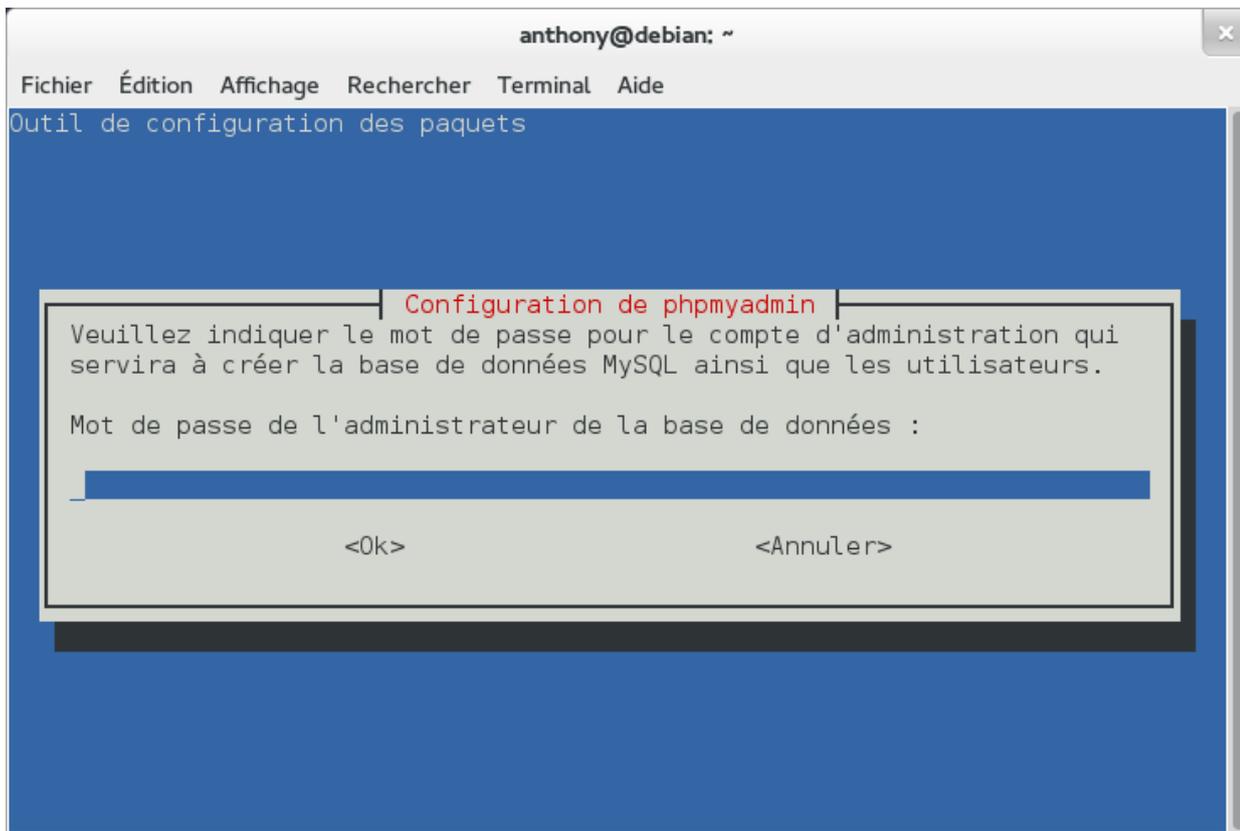
Associez ensuite MySQL à Apache 2 en cochant la case **Apache 2** (à l'aide de la touche Espace), puis, avec la touche Tabulation, rendez-vous sur **Ok** puis validez.



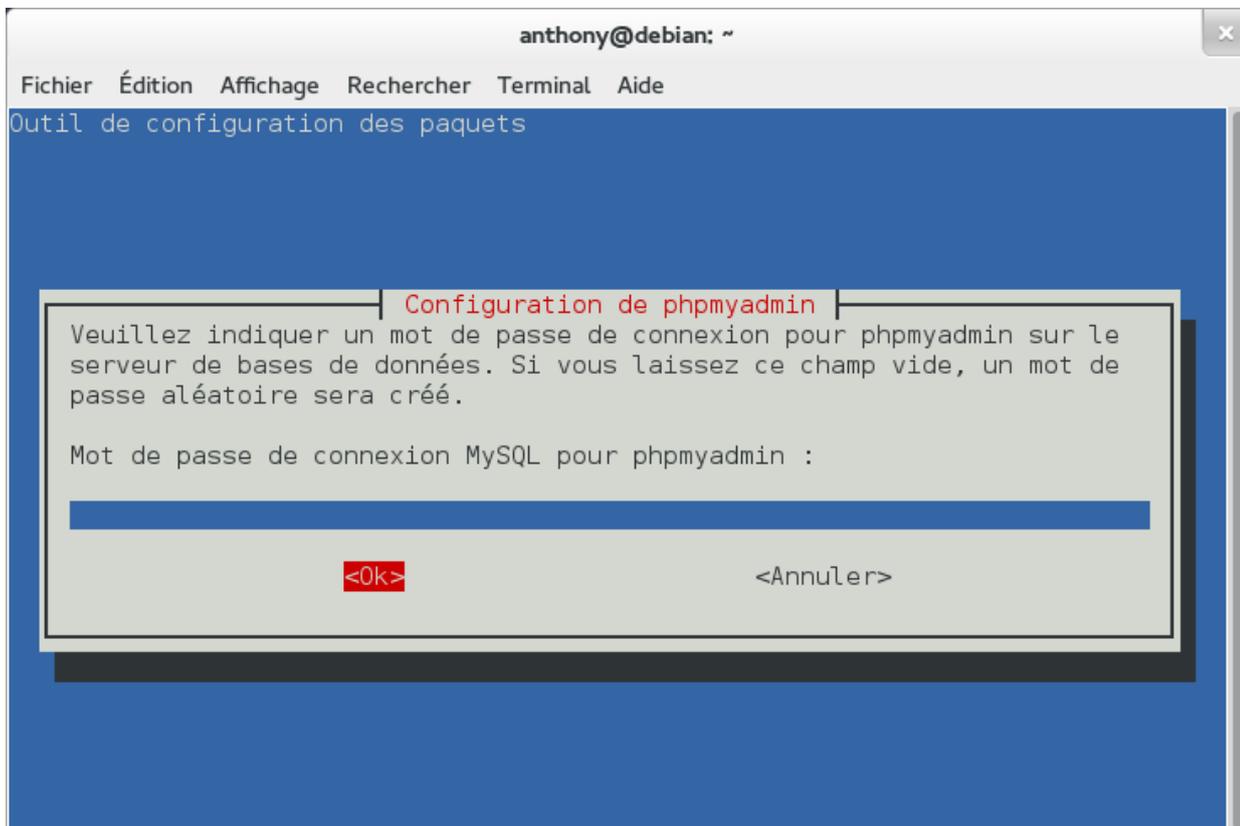
Vous devez ensuite configurer PHPMYAdmin. Je vous conseille de répondre **Oui** à la question **Faut-il configurer la base de données de phpmyadmin avec dbconfig-common** afin de bénéficier d'une configuration faite rapidement par l'assistant :



Indiquez le mot de passe du compte MySQL administrateur de la base de données.



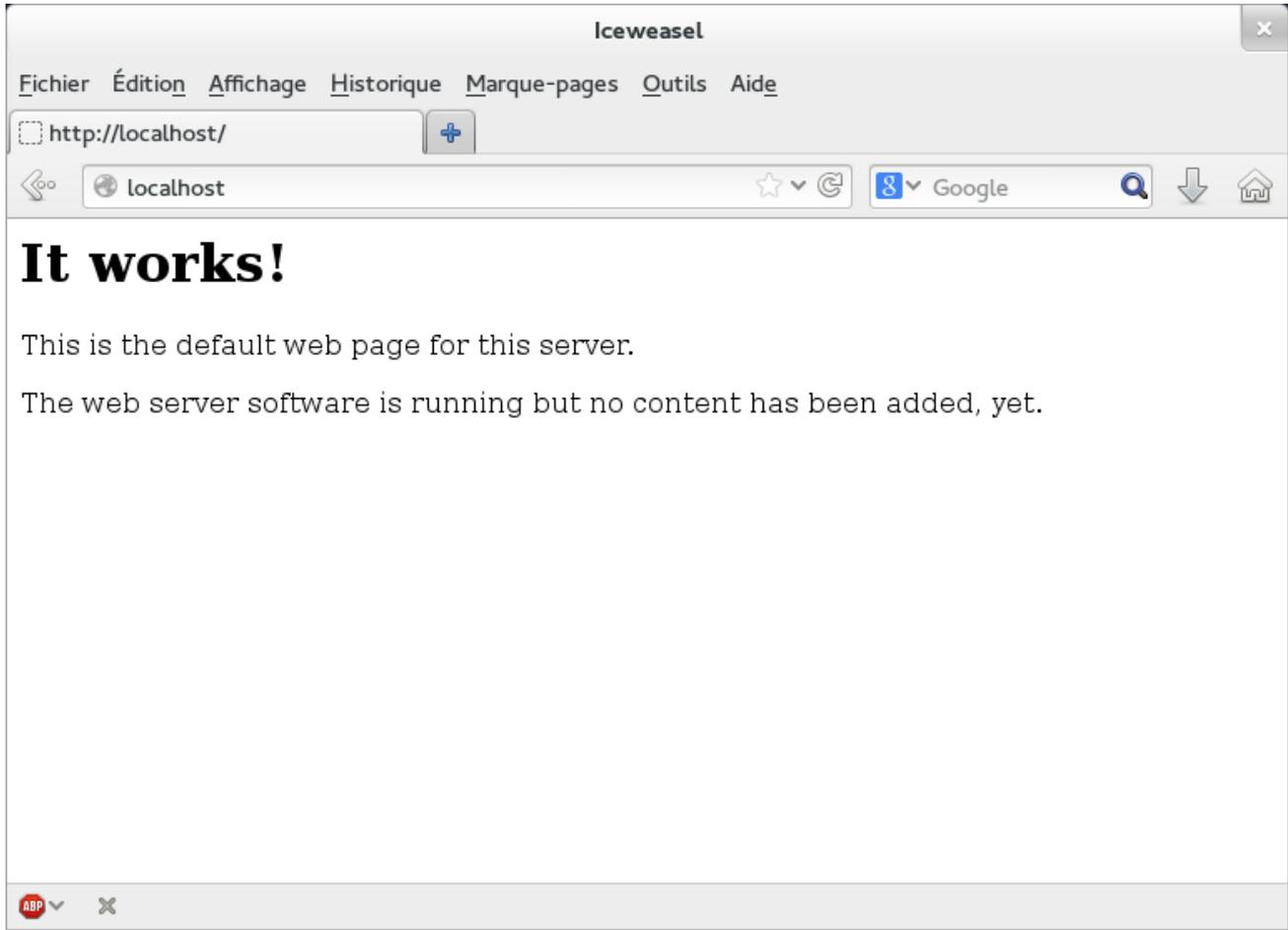
Indiquez ensuite le mot de passe du compte administrateur de PHPMysql :



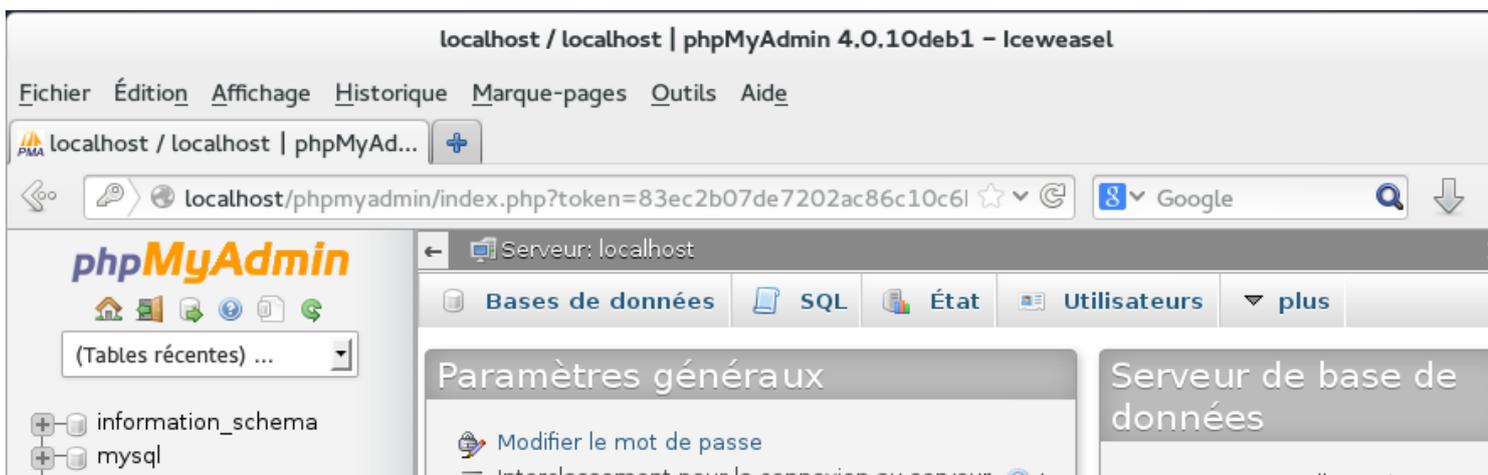
A ce stade, notre serveur Web LAMP doit être fonctionnel. Ouvrez un navigateur web, puis rendez-vous à l'adresse IP de votre serveur (pour ce tutorial, j'effectue les tests en localhost, mon adresse est donc http://localhost). Vous devriez avoir un message **It Works** indiquant qu'Apache 2 est fonctionnel.

Il vous faut maintenant activer le module PHP pour Apache (Dotdeb l'avait intégré par défaut mais ce n'est plus le cas).

```
~  
#  
a2enmod php5~  
#  
service apache2 restart
```



Rajoutez ensuite à la fin de l'adresse web la mention /phpmyadmin (ce qui donne chez moi http://localhost/phpmyadmin) et constatez le bon fonctionnement non seulement de PHP (PHPMyAdmin a besoin de PHP pour fonctionner) mais également de MySQL en entrant vos informations de connexion (compte : **root** avec votre mot de passe root MySQL) :



Que faire si cela ne fonctionne pas ? Au lancement d'Apache, vous pouvez obtenir le message suivant :

Apache is running a threaded MPM, but your PHP Module is not compiled to be threadsafe. You need to recompile PHP.

Si vous obtenez ce message, vous devez installer Apache en mode MPM, avec la commande suivante :

~

#

apt-

get install apache2-

mpm-

prefork

Test du bon fonctionnement d'Apache, PHP 5.5, MySQL 5.5 et PHPMyAdmin

Nous allons maintenant vérifier la version de PHP ainsi que les modules installés par défaut. Pour ce faire, rendez-vous dans le répertoire **/var/www** et créez un fichier intitulé **test.php** avec nano :

~

#

nano /

var/

www/

test.

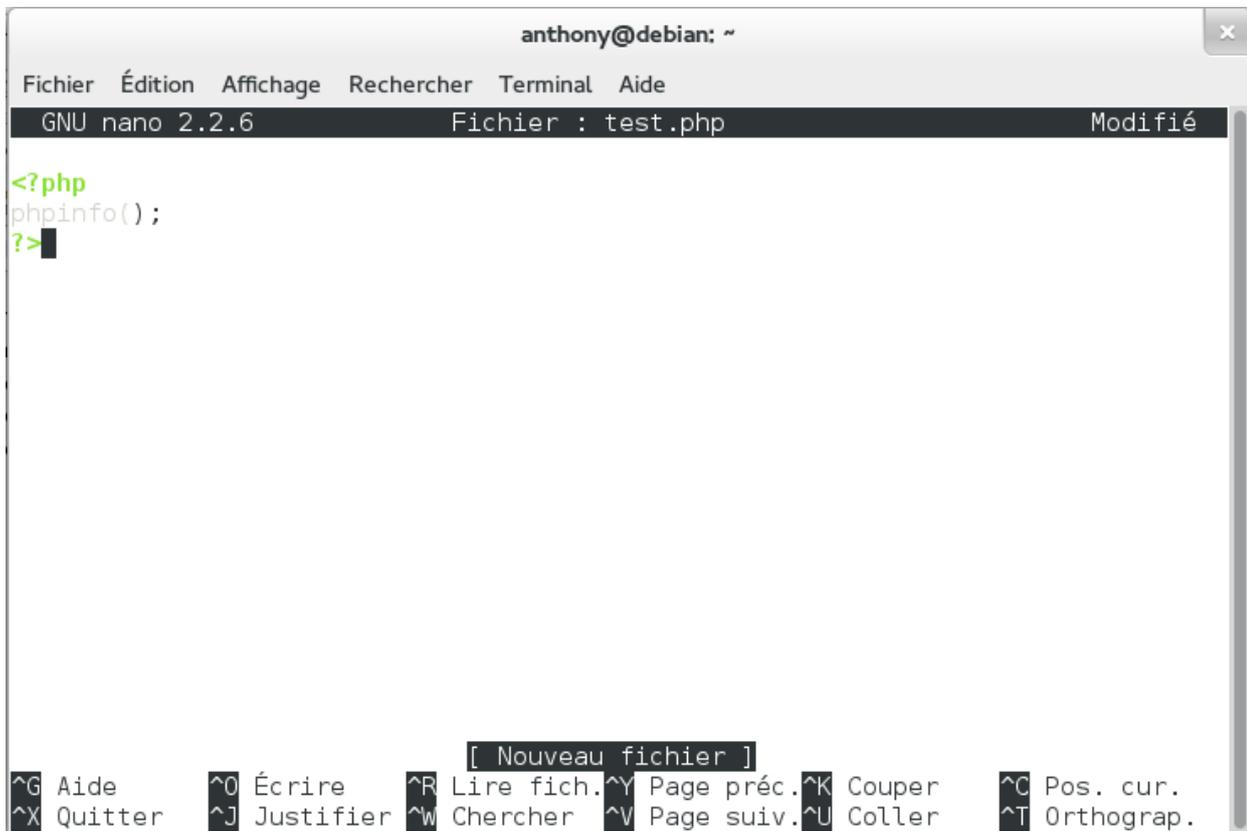
php

Insérez les lignes suivantes puis sauvegardez le fichier (**CTRL + X** puis **O** et **Entrée**).

```
<?php
```

```
phpinfo();
```

```
?>
```

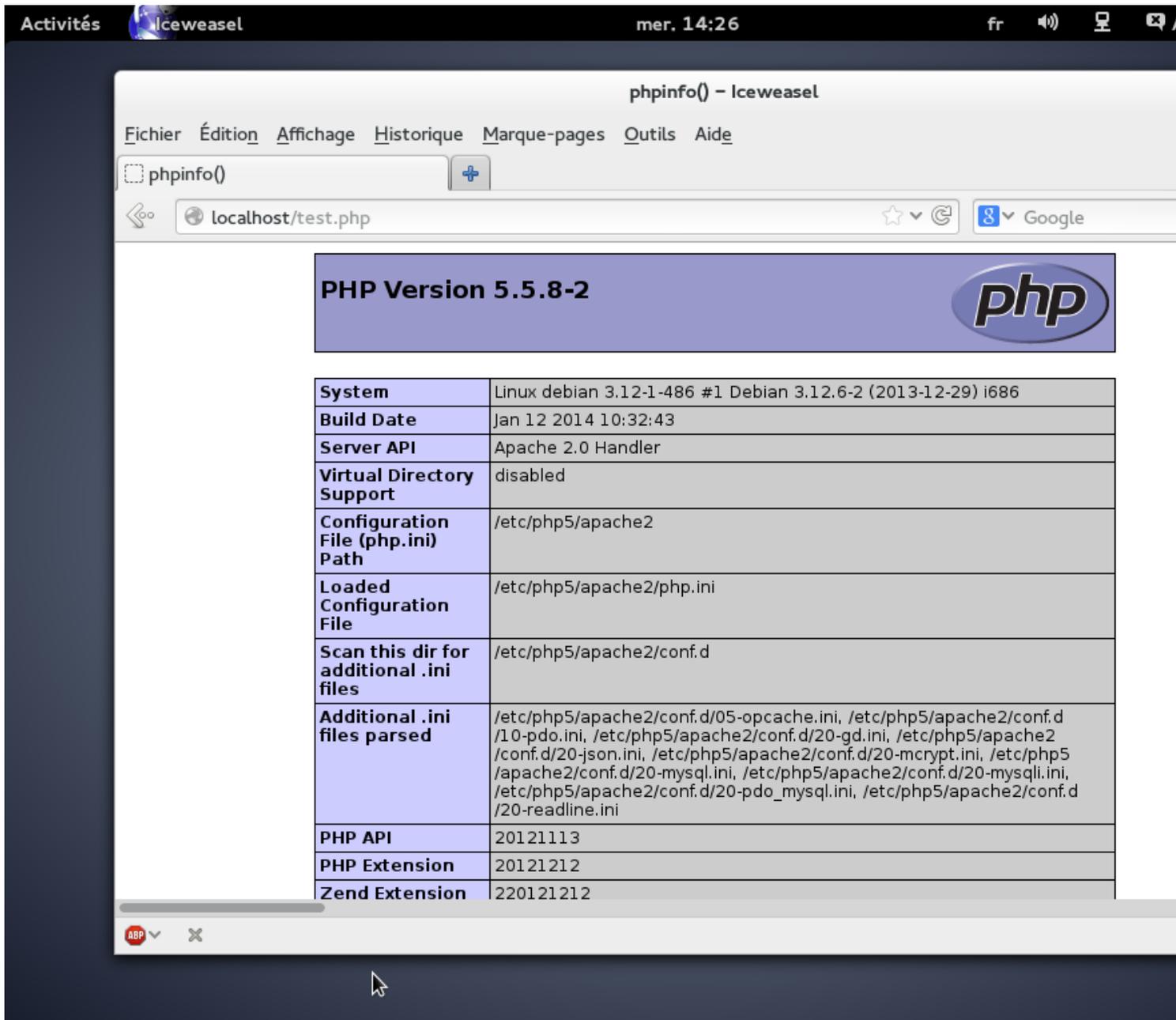


The screenshot shows a terminal window titled 'anthony@debian: ~' with a nano text editor open. The editor's status bar at the top indicates 'GNU nano 2.2.6' and 'Fichier : test.php'. The main editing area contains the following PHP code:

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

The cursor is positioned at the end of the third line. At the bottom of the window, a menu bar is visible with various keyboard shortcuts for navigation and editing, such as '^G Aide', '^X Quitter', '^O Écrire', '^J Justifier', '^R Lire fich.', '^W Chercher', '^Y Page préc.', '^V Page suiv.', '^K Couper', '^U Coller', '^C Pos. cur.', and '^T Orthograp.'. A '[Nouveau fichier]' message is also present above the menu bar.

Prenez un navigateur web, rendez-vous à l'adresse IP de votre serveur puis indiquez la page **test.php** (pour ce tutoriel, cela donne **http://localhost/test.php**). Si tout se passe bien, vous devriez avoir un tableau ressemblant à celui-ci et indiquant non seulement la version de PHP installée, mais aussi les différents modules chargés :



A ce stade, vous disposez d'un serveur web fonctionnel pour effectuer la plupart des tâches de base. Nous allons maintenant voir comment installer d'autres choses qui peuvent vous être importantes.

Installation de sendmail

Pour [envoyer des mails avec PHP](#), vous aurez besoin d'installer a minima sendmail. Pour ce faire, exécutez les commandes suivantes :

~
#

```
apt-
get install sendmail-
bin~
#
```

```
apt-
get install sendmail
```

Installation de postgresSQL et PHPPgAdmin

PostgreSQL est un serveur de gestion de bases de données relationnelles (SGBDR) Open Source et performant, bien plus que MySQL dans des cas d'utilisation avancés. Vous pourriez en avoir besoin. Nous installerons également PHPPgAdmin, l'équivalent de PHPMyAdmin mais adapté aux bases de données PostgreSQL.

```
~
#
apt-
get install php5-
pgsql
```

Répondez par l'affirmative (touche **O**) aux messages éventuels d'installation de dépendances.

Il vous faut maintenant installer le serveur PostgreSQL proprement dit, à l'aide des commandes suivantes :

```
~
#
apt-
get install postgresql postgresql-
client phppgadmin
```

Répondez par l'affirmative (touche **O**) aux messages éventuels d'installation de dépendances.

Nous allons maintenant créer l'alias qui va permettre d'accéder à PHPPgAdmin depuis votre navigateur web. Pour ce faire, ouvrez le fichier **apache2.conf**.

```
~
#
nano /
etc/
apache2/
apache2.
conf
```

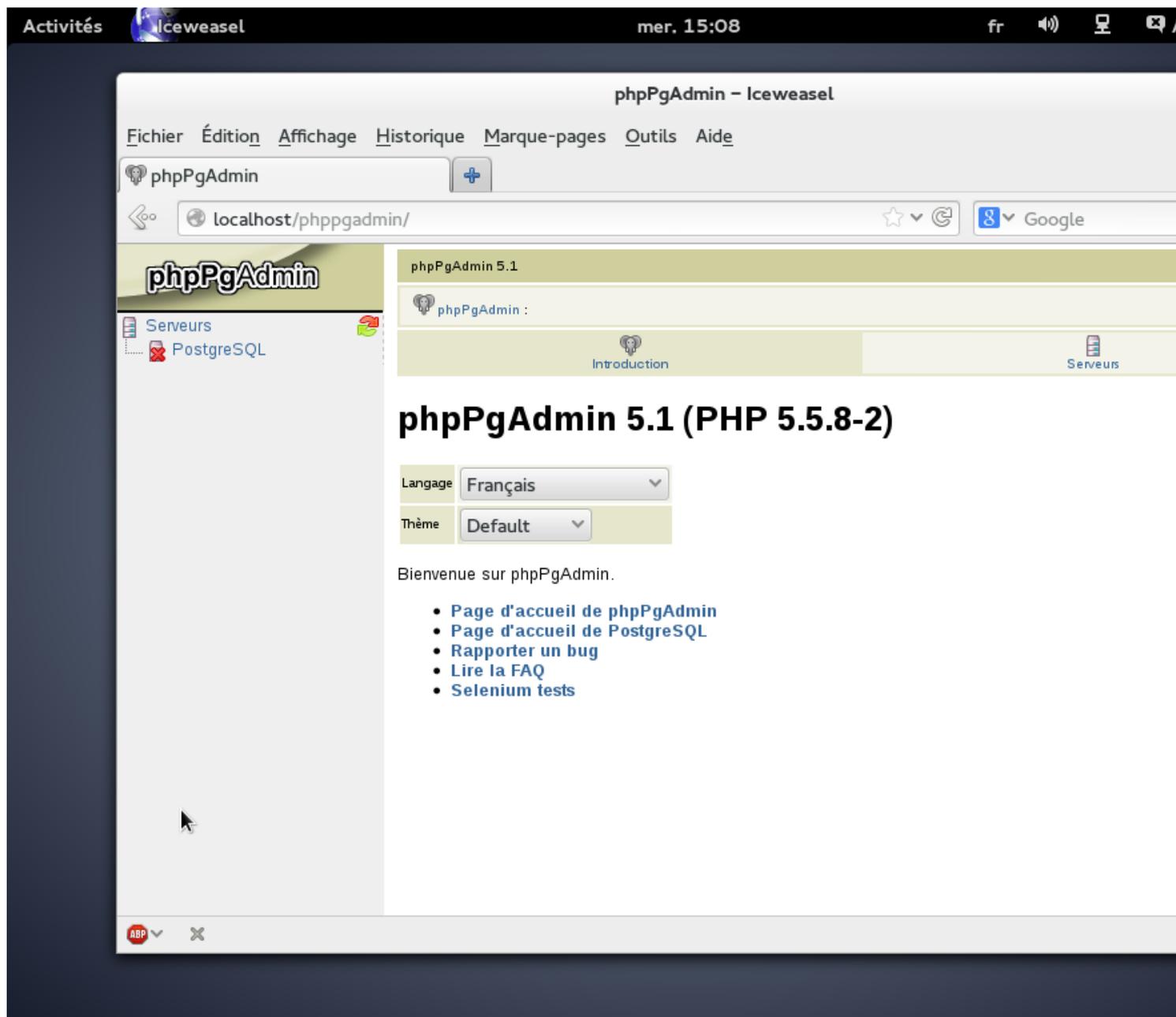
Rajoutez tout en bas du fichier apache2.conf la ligne suivante : **Include /etc/apache2/conf.d/phppgadmin**. Ceci aura pour effet d'inclure le fichier d'alias déjà créé par l'installation de phppgadmin.

Par défaut, il se peut que l'utilisation de PHPPgAdmin soit restreinte à la machine locale, il vous sera alors impossible d'y accéder à distance (ce qui, dans notre cas, est plutôt problématique. Ouvrez le fichier **/etc/apache2/conf.d/phppgadmin** et commentez la ligne **Allow from 127.0.0.1[...]** en rajoutant un **#** devant. Décommentez (en enlevant le signe **#** la ligne **Allow From All**. Sauvegardez le fichier (et relancez Apache ou continuez ce tutoriel).

Vous devez ensuite redémarrer le serveur Apache :

```
~
#
/
etc/
init.
d/
apache2 restart
```

Prenez votre navigateur web, rendez-vous cette fois à l'adresse de votre serveur, suivi du répertoire **phpPgAdmin**. Cela donne chez moi : <http://localhost/phpPgAdmin/>. Si l'installation s'est bien passée, vous devriez vous retrouver devant PHPPgAdmin :



Il faut maintenant se connecter au serveur PostgreSQL, puis créer un utilisateur et une base de données. Pour ce faire, dans un terminal, tapez la commande **su - postgres**, validez par Entrée, puis tapez la commande **psql**. Vous vous retrouvez dans postgresql.

```
~  
#  
su -  
postgres~  
#  
psqlpsql (
```

9.3

.
2
)
Saisissez "
help
"

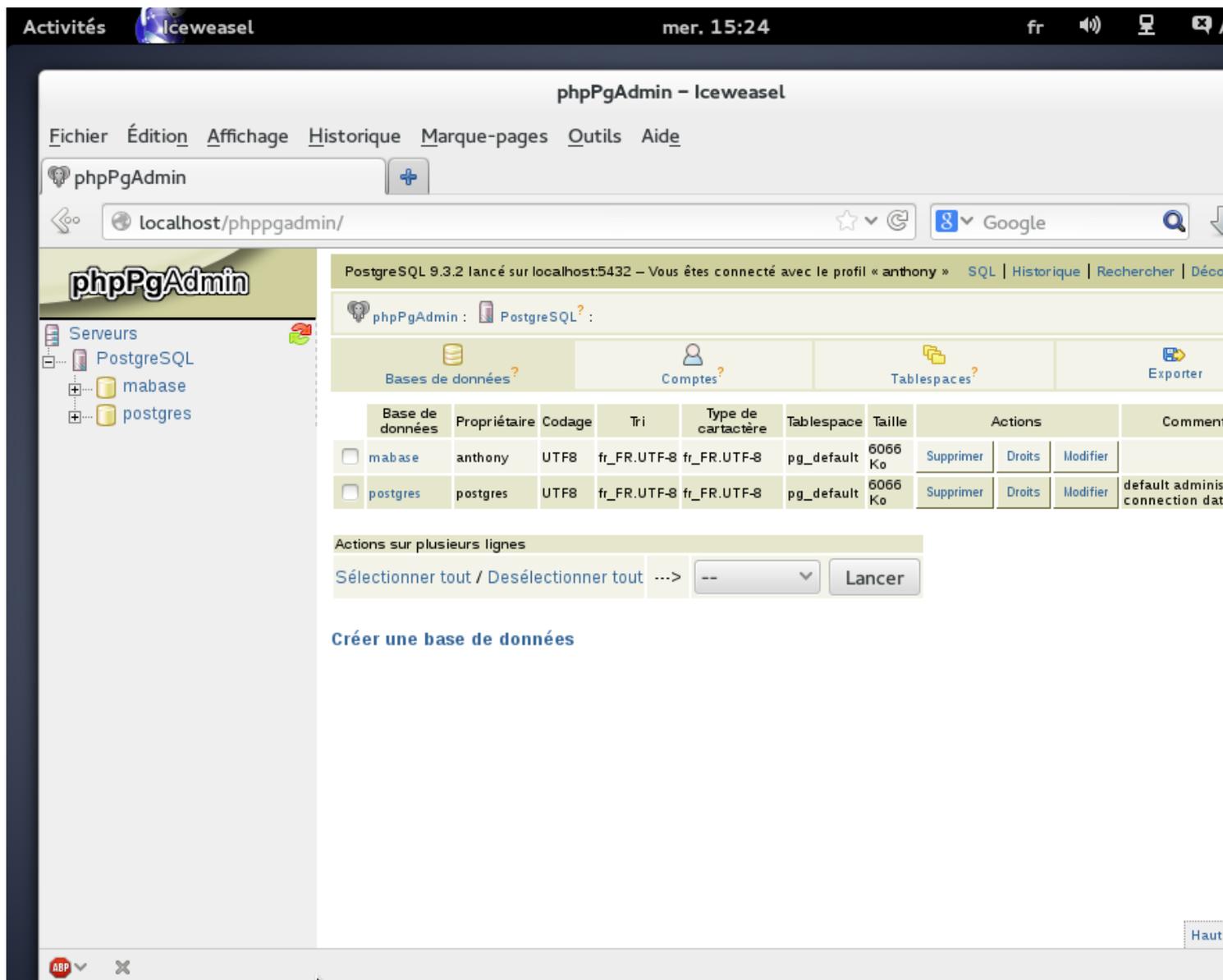
pour l'aidepostgres=
#

Pour créer l'utilisateur **anthony** (ayant le mot de passe **anthony** et une base nommée **mabase**, exécutez les commandes suivantes (après chaque point-virgule, appuyez sur Entrée puis entrez la commande suivante) :

```
postgres=  
#  
CREATE USER anthony WITH PASSWORD 'anthony'  
;  
postgres=  
#  
CREATE DATABASE mabase OWNER anthony;
```

Quittez le serveur PostgreSQL avec la commande `\q` puis validez par Entrée.

Dans PHPPgAdmin, cliquez en haut à gauche sur PostgreSQL, puis renseignez dans le formulaire de connexion (à droite) les informations de l'utilisateur que vous venez de créer. Vous devriez être connecté à votre base de données !



Activation du mod_rewrite d'apache

Le mod_rewrite d'Apache permet de faire de la redirection d'URL (aussi appelée [URL Rewriting](#)) et va vous permettre de faire croire à l'internaute qu'il se trouve sur une page X.php alors que la page réelle est par exemple Z.php?param=X. Cela vous permet d'améliorer le référencement dans les moteurs de recherche et augmente légèrement la sécurité de vos sites Web. Exécutez la commande suivante dans un terminal :

```
~
#
a2enmod rewrite~
#
/
etc/
init.
d/
apache2 restart
Installation de PEAR
```

PEAR est un ensemble de packages et bibliothèques très utiles. Pour installer PEAR, il vous suffit d'exécuter la commande suivante :

```
~
#
apt-
get install php-
pear
Installation d'APC pour PHP
```

APC est un accélérateur pour PHP. Il consiste à mettre en cache les informations d'OP-Code. PHP étant un langage interprété, il est analysé à chaque fois que vous appelez la page web, puis transformé en OP-Code. Pour faire simple, APC va mettre en cache l'OP-Code pour gagner le temps de cette transformation. Les gains peuvent être très importants, et c'est gratuit !

A l'heure où j'écris ces lignes, APC n'est pas disponible pour PHP 5.5. Lorsque ce sera le cas, ouvrez un terminal root et exécutez les commandes suivantes :

```
~
#
apt-
get install php5-
apc~
#
/
etc/
init.
d/
apache2 restart
```

Ce tutoriel touche à sa fin. Vous venez d'installer un serveur web LAMP sous Debian Wheezy. Bonne administration !

Source : <http://www.vulgarisation-informatique.com/installer-serveur-lamp-debian.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)