



par Sascha Blum (homepage)

Installer un système LAMP



L'auteur:

J'aime vraiment utiliser Linux en raison de sa grande puissance et de sa stabilité et surtout pour toutes les ressources et options qu'il propose. Encore mieux, il est ouvert à tout le monde et ainsi chacun peut s'investir dans son développement.

Traduit en Français par:

Georges Tarbouriech
<gt(at)linuxfocus.org>

Résumé:

Dans ce tutoriel, j'aimerais montrer comment installer un serveur Linux possédant toutes les caractéristiques utiles. En d'autres termes, je vais vous expliquer comment installer un système LAMP.

Mais d'abord, je vais vous dire ce que signifie l'abréviation LAMP. LAMP veut dire Linux Apache MySQL PHP. Comme vous pouvez le deviner d'après le nom, un système LAMP consiste en un système d'exploitation Linux, un serveur web Apache, une base de données MySQL et le langage de script PHP.

Introduction

Ce tutoriel explique comment installer un système LAMP avec des Objets Dynamiques Partagés (Dynamic Shared Objects ou DSO).

Le principal avantage des DSO par rapport à une installation statique vient de ce que vous pouvez facilement remplacer chaque module par une version plus récente sans devoir recompiler ou réinstaller tous les autres modules. Peu importe qu'il s'agisse du module PDF-Lib, GD-Lib ou autre. Avec une installation statique, si par exemple, vous souhaitez mettre à jour PHP 4.2.3 vers PHP 4.2.4, vous devez tout recompiler et réinstaller - et par "tout", j'entends le serveur Apache, la GD-Lib, la PDF-Lib et tous les autres modules requis (et bien sûr, PHP proprement dit). Avec une installation DSO, seul PHP sera affecté et tout le reste sera maintenu en l'état.

Note: il est recommandé de lire le fichier README de chaque paquetage avant de l'installer ou de le compiler, puisque chaque installation est spécifique. Souvent, une installation fructueuse dépend d'une option ou d'une autre que vous pouvez ou devez définir dans ./configure. Cela dit, selon mon expérience, cette installation devrait fonctionner du premier coup. Si vous obtenez des erreurs, consultez le fichier README. Vous devez avoir un accès root pour l'installation ! Assez de préambule. Commençons notre installation du système LAMP.

Lisez attentivement ce tutoriel dans sa totalité avant de commencer l'installation !

Le nécessaire et le téléchargement des sources

Les paquetages suivants sont requis et vous devez les télécharger avant de commencer l'installation :

- Apache 1.3.27
(<http://www.apache.org/>)
Téléchargement direct :
http://www.apache.org/dist/httpd/apache_1.3.27.tar.gz (2,2 MB)
- MySQL
(<http://www.mysql.org/>)
Paquetages RedHat (rpm) :
MySQL 3.23.52 Server (i386) (7.4M)
MySQL 3.23.52 programmes Client (i386) (2.2M)
MySQL 3.23.52 Bibliothèques et fichiers en-têtes pour le développement (i386) (743K)
MySQL 3.23.52 bibliothèques partagées Client (i386) (232K)
- zlib 1.1.4
(<http://www.gzip.org/zlib/>)
Téléchargement :
<ftp://ftp.info-zip.org/pub/infozip/zlib/zlib-1.1.4.tar.gz> (177 KB)
- GD Library 1.8.4
(<http://www.boutell.com/gd/>)
Téléchargement :
<http://www.boutell.com/gd/http/gd-1.8.4.tar.gz> (252 KB)
Note: pour des raisons de licence, la bibliothèque GD ne supporte plus le format GIF (et ce depuis un certain temps !)
- PDF Lib 4.0.3
(<http://www.pdflib.com/pdflib/index.html>)
Téléchargement :
<http://www.pdflib.com/pdflib/download/pdflib-4.0.3-Linux.tar.gz> (3,2 MB)
- PHP 4.2.3
(<http://www.php.net/>)
Téléchargement :
http://us3.php.net/do_download.php?download_file=php-4.2.3.tar.gz (3,3 MB)

Installation

Une fois ces paquetages téléchargés, vous êtes prêts. Copiez d'abord les fichiers dans le répertoire suivant (si vous ne l'avez pas déjà fait, créez le répertoire lamp par la commande mkdir /usr/local/src/lamp):

```
/usr/local/src/lamp/
```

Les seuls fichiers à ne pas copier sont les rpm de MySQL. Ils peuvent être installés directement de la manière habituelle. Le mieux est de commencer par là. Pour la marche à suivre, voir la section "MySQL 3.23.52" ci-dessous.

Six paquetages au format tar.gz devraient se trouver dans le répertoire /usr/local/src/lamp/. Vous devez maintenant les décompresser. Procédez comme indiqué ci-dessous.

Note: les commandes à taper apparaissent en caractères gras; L'affichage des sorties est en caractères normaux. Chaque entrée est précédée d'un symbole >.

Ouvrez une console texte (un terminal shell, e.g. Bash), et exécutez les commandes suivantes :

```
user:~ ➤su  
[Entrez votre mot de passe root]  
root: ~➤cd /usr/local/src/lamp  
root:/usr/local/src/lamp > tar -xvzf apache_1.3.27.tar.gz  
root:/usr/local/src/lamp > tar -xvzf zlib-1.1.4.tar.gz  
root:/usr/local/src/lamp > tar -xvzf libpng-1.2.2.tar.gz  
root:/usr/local/src/lamp > tar -xvzf gd-1.8.4.tar.gz  
root:/usr/local/src/lamp > tar -xvzf pdflib-4.0.3.tar.gz  
root:/usr/local/src/lamp > tar -xvzf php-4.2.3.tar.gz
```

Après avoir décompressé tous les paquetages, tapez la commande "ls -l" pour afficher tous les répertoires.

A partir de là, il est essentiel de suivre les étapes d'installation exactement dans l'ordre présenté. Ceci est dû au fait que certains paquetages sont dépendants d'autres pour fonctionner correctement. Par exemple, la bibliothèque GD a besoin de zlib et libpng, qui à son tour, a besoin de zlib. Passons maintenant au serveur web Apache.

Apache 1.3.27

Note: Lisez le fichier README ! Il existe de nombreuses options susceptibles d'être passées à ./configure.

Ne compilez jamais le serveur Apache avec l'option --enable-module=all ! Si vous le faites, rien ne

fonctionnera. Le mieux consiste à spécifier le moins de modules possible. Normalement, c'est plus que suffisant pour le support de DSO. Vous pourrez ensuite ajouter autant de modules que vous le souhaitez; après tout, c'est le principal intérêt d'une installation DSO.

Pour installer et configurer Apache, procédez comme suit.

Ouvrez une console texte (un terminal shell, e.g. Bash), comme précédemment.

Note: ne tapez pas `user:/usr/local/src/lamp >` dans vos commandes. Il s'agit du prompt et c'est la manière pour Linux de vous dire qu'il attend une entrée. Votre prompt peut avoir un autre aspect puisqu'il peut être configuré.

```
user:/usr/local/src/lamp > cd apache_1.3.27
user:/usr/local/src/lamp/apache_1.3.27 > su
[Entrez votre mot de passe root]
root:/usr/local/src/lamp/apache_1.3.27 > ./configure --prefix=/usr/local/apache/1.3.27
--datadir=/web/htdocs --enable-rule=SHARED_CORE --enable-module=so
```

Note: tapez cette dernière partie sur une ligne ! Habituellement, il existe un espace avant le `--`. Le répertoire document dans lequel vos sites seront stockés ultérieurement vient après `-datadir`. Vous pouvez bien sûr choisir un autre emplacement. Toutefois, si vous utilisez un répertoire différent de `"/web/htdocs"`, n'oubliez pas de modifier les chemins en conséquence.

```
root:/usr/local/src/lamp/apache_1.3.27 > make
root:/usr/local/src/lamp/apache_1.3.27 > make install
```

Si vous avez tapé tout ceci correctement, votre serveur Apache devrait être compilé et installé.

MySQL 3.23.52

Si vous avez suivi les instructions précédentes de ce tutoriel, tout est déjà installé.

Note de sécurité : si votre serveur est connecté à un réseau public, c'est-à-dire un intranet ou Internet, affectez un mot de passe aussi complexe que possible à l'utilisateur root du serveur MySQL !

```
root:/home/user/download/mysql > rpm -Uvh MySQL-3.23.52-1.i386.rpm
root:/home/user/download/mysql > rpm -Uvh MySQL-client-3.23.52-1.i386.rpm
root:/home/user/download/mysql > rpm -Uvh MySQL-devel-3.23.52-1.i386.rpm
root:/home/user/download/mysql > rpm -Uvh MySQL-shared-3.23.52-1.i386.rpm
```

Note: remplacez `/home/user/download/mysql` par le répertoire dans lequel se trouvent les fichiers RPM correspondants.

zlib 1.1.4

```
root:/usr/local/src/lamp/apache_1.3.27 > cd /zlib-1.1.4/  
root:/usr/local/src/lamp/zlib-1.1.4 > ./configure --shared  
root:/usr/local/src/lamp/zlib-1.1.4 > make  
root:/usr/local/src/lamp/zlib-1.1.4 > make install
```

Commentaire : nous utilisons l'option --shared pour indiquer à zlib que nous voulons inclure la bibliothèque en tant que module dynamique dans PHP.

libpng 1.2.3

L'installation de libpng est un peu différente de la procédure habituelle. Positionnez-vous dans le répertoire /libpng-1.2.3/scripts/ :

```
root:/usr/local/src/lamp/zlib-1.1.4 > cd ../libpng-1.2.3/scripts/
```

Tapez ensuite les commandes suivantes :

```
root:/usr/local/src/lamp/libpng-1.2.3/scripts > cp makefile.linux ../makefile  
root:/usr/local/src/lamp/libpng-1.2.3/scripts > cd ..
```

Par ces commandes vous avez simplement copié le fichier make concerné dans le répertoire principal de libpng. Maintenant vous devez lire le fichier makefile et le modifier en fonction de votre système, par exemple, si vous avez des répertoires d'en-têtes particuliers. Normalement toutes les données du fichier sont correctes, mais vérifiez quand même, ça vous aidera à détecter les erreurs plus rapidement.

Pour continuer, tapez la commande suivante :

```
root:/usr/local/src/lamp/libpng-1.2.3 > make test
```

Si vous n'avez aucun message d'erreur, vous pouvez installer libpng par la commande suivante :

```
root:/usr/local/src/lamp/libpng-1.2.3 > make install
```

gd-1.8.4

Positionnez-vous dans le répertoire gd-1.8.4:

```
root:/usr/local/src/lamp/libpng-1.2.3 > cd ../gd-1.8.4/
```

Ici aussi, vous devriez vérifier le makefile. Si quelque chose a été changé dans votre système, faites les modifications correspondantes dans le fichier makefile dès maintenant. Vous pouvez lire et éditer le fichier avec l'éditeur de texte de votre choix.

Normalement, vous ne devriez pas avoir à modifier le fichier makefile.

S'il vous satisfait, tapez la commande suivante :

```
root:/usr/local/src/lamp/gd-1.8.4 > make  
root:/usr/local/src/lamp/gd-1.8.4 > make install
```

Si vous obtenez des erreurs, tapez ce qui suit :

```
root:/usr/local/src/lamp/gd-1.8.4 > make clean
```

Ne tapez cette commande qu'en cas d'erreurs ! Si vous exécutez un make clean, vous devrez revérifier le makefile et l'adapter en conséquence et ensuite relancer le make.

Note: vérifiez les définitions INCLUDEDIRS et LIBDIRS!

PDF-Lib 4.0.3

C'est un peu plus simple puisque le module est déjà compilé. Il suffit de le copier dans le répertoire /usr/local/lib.

Pour ce faire, tapez ce qui suit :

```
root:/usr/local/src/lamp/gd-1.8.4 > cd /  
root:/ > cp /usr/local/src/lamp/pdflib-4.0.3-Linux/bind/php/php-4.2.1/libpdf_php.so /usr/  
local/lib/libpdf_php.so
```

PHP 4.2.3

Enfin, vous devez installer PHP.

Positionnez-vous dans le répertoire PHP :

```
root:/ > cd /usr/local/src/lamp/php-4.2.3/  
root:/usr/local/src/lamp/php-4.2.3 > ./configure --with-apxs=/usr/local/apache/1.3.27/bin/apxs  
--enable-track-vars --enable-ftp --with-zlib --with-gd --with-sockets --enable-sockets
```

```
--with-sysvshm --with-sysvsem --disable-debug --with-pdflib-dir=/usr/local/lib
--with-tiff-dir=/usr/local/lib --with-jpeg-dir=/usr/local/lib --with-png-dir=/usr/local/lib
--with-zlib-dir=/usr/local/lib --with-mysql --with-xml
```

Note: tapez cette dernière partie sur une seule ligne ! Habituellement, il y a un espace avant le --. Il n'y a pas d'erreurs dans les lignes 2 et 3 ("sysvshm" et "sysvsem").

Comme précédemment, tapez ce qui suit :

```
root:/usr/local/src/lamp/php-4.2.3 > make
root:/usr/local/src/lamp/php-4.2.3 > make install
```

Note: compiler (make) PHP peut durer un certain temps sur des systèmes peu performants. Ne soyez donc pas impatients s'il semble ne rien se produire pendant de longs moments. Vous pouvez effacer le répertoire /usr/local/src/lamp (en tant que root) par "rm -r /usr/local/src/lamp". Soyez attentifs en tapant ceci, sinon, en exécutant un "rm -r /" en tant que root, vous détruirez la totalité du système. Toutefois, sachez que si vous effacez "/usr/local/src/lamp", il sera plus compliqué de désinstaller ou de mettre à jour le système. Il serait donc préférable de détruire uniquement les paquetages ".tar.gz" et de conserver les répertoires contenant les sources.

Configuration

httpd.conf

Voilà pour l'installation. Maintenant, voyons la configuration.

Tout d'abord, nous devons dire au serveur Apache ce qu'il doit faire avec les fichiers *.php- ou *.php3.

Positionnez-vous dans le répertoire "conf" d'Apache :

```
root:/usr/local/src/lamp/php-4.2.3 > cd /usr/local/apache/1.3.27/conf
root:/usr/local/apache/1.3.27/conf >
```

Ouvrez ensuite le fichier "httpd.conf" dans un éditeur de manière à l'éditer et à le sauvegarder.

Note: l'éditeur "Kate" est pratique pour éditer le fichier de configuration. KDE doit bien sûr être actif. Pour le lancer, pressez Alt + F2 => kdesu kate. Pressez Ctrl + G pour atteindre la ligne de votre choix.

Vers la ligne 190 de ce fichier vous trouverez :

```
#
```

```
#Dynamic Shared Object (DSO) Support
#
#To be able to use the functionality of a module which was built as a DSO you
#have to place corresponding 'LoadModule' lines at this location so the
#directives contained in it are actually available _before_ they are used.
#Please read the file README.DSO in the Apache 1.3 distribution for more
#details about the DSO mechanism and run 'httpd -l' for the list of already
# built-in (statically linked and thus always available) modules in your httpd
#binary.
```

Tapez ce qui suit si ça n'existe pas déjà :

```
LoadModule php4_module libexec/libphp4.so
```

Vers la ligne 770, vous trouverez ceci :

```
#AddType allows you to tweak mime.types without actually editing it, or to
#make certain files to be certain types.
#AddType application/x-tar .tgz
```

Ajoutez ce qui suit :

```
AddType application/x-httpd-php .htm
AddType application/x-httpd-php .html
AddType application/x-httpd-php .phtm
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php .php
AddType application/x-httpd-php .php3
AddType application/x-httpd-php .php4
AddType application/x-httpd-php-source .phps
```

Note: tapez exactement ceci, sinon vous aurez des erreurs.

Si vous ne voulez pas que l'analyseur (parser) PHP exécute les fichiers HTML, vous pouvez ignorer les lignes suivantes :

```
AddType application/x-httpd-php .htm
AddType application/x-httpd-php .html
```

Le fichier httpd.conf est maintenant correct.

php.ini

Vous devez maintenant définir et probablement adapter le fichier

<p>Qu'est-ce qu'un analyseur (parser) ?</p>
--

vous devez maintenant définir et probablement adapter le fichier php.ini.

Vous devez d'abord copier le fichier php.ini au bon endroit. Positionnez-vous dans le répertoire d'installation de PHP :

```
root:/usr/local/apache/1.3.27/conf > cd  
/usr/local/src/lamp/php-4.2.3/
```

Copiez le fichier "php.ini-dist" dans le répertoire /usr/local/lib et renommez-le "php.ini". Faites-le comme suit:

```
root:/usr/local/src/lamp/php-4.2.3 > cp php.ini-dist  
/usr/local/lib/php.ini
```

Ajoutez ensuite "pdflib" en tant qu'extension dans le fichier php.ini. Ainsi, PHP sait quoi faire des fonctions PDF correspondantes si vous en avez besoin. PHP trouve les autres modules tout seul (zlib, GD, etc.).

Ouvrez le fichier "/usr/local/lib/php.ini" dans un éditeur. La section des extensions se situe vers la ligne 371.

Elle devrait ressembler à quelque chose de ce genre :

```
;Directory in which the loadable extensions (modules) reside.  
....
```

extension_dir = ./ <= supprimez ceci et remplacez-le par :

```
extension_dir = /usr/local/lib  
extension=libpdf_php.so
```

Enregistrez le fichier.

C'est terminé - vous avez maintenant un système LAMP complet et fonctionnel !

Maintenant testons le serveur. Voyons s'il démarre correctement. La première étape consiste à vérifier qu'aucun serveur n'est encore actif (si un serveur web a été installé avec la distribution, par exemple). Pour cela, tapez ce qui suit :

```
root:/usr/local/src/lamp/php-4.2.3 > killall httpd
```

Un analyseur est un logiciel capable d'interpréter du texte. Ce texte peut être du code source (C++ par exemple) ou un document contenant des balises (HTML par exemple). L'analyseur cherche les éventuelles erreurs syntaxiques ou sémantiques du texte et transmet le texte analysé à l'application concernée, généralement sous forme de code interne efficace et compact.

Le texte filtré par un analyseur prend un peu plus de temps à s'afficher. Ca signifie que des pages de véritable HTML s'affichent plus vite que des pages PHP ou des scripts. Toutefois, l'utilisateur ne remarque pas vraiment la différence. Une certaine lenteur peut être constatée si plusieurs utilisateurs accèdent à la même chose : si plusieurs utilisateurs appellent une page PHP ou un script en même temps, l'affichage peut devenir plus lent selon le matériel concerné. Par conséquent, si vous souhaitez rendre public votre système LAMP, c'est-à-dire en le connectant à Internet, un intranet ou un réseau, prévoyez une machine puissante et rapide, sinon le système sera très lent. Si d'un autre côté, vous souhaitez utiliser votre système LAMP pour développer des pages PHP ou des scripts relatifs à une base de données MySQL, vous pouvez sans crainte vous servir d'un vieux PC ou d'un portable. Ceci s'applique également si vous êtes la seule personne à exécuter ou afficher des pages PHP ou des scripts sur votre machine.

Essayons de démarrer le nouveau serveur :

```
root:/ > /usr/local/apache/1.3.27/bin/apachectl start
```

Si vous voyez ce message...

```
/usr/local/apache/1.3.27/bin/apachectl start: httpd started
```

... tout va bien et le serveur est opérationnel et actif !

Positionnez-vous dans le répertoire "web/htdocs" (DocumentRoot - si vous avez donné un nom différent, n'oubliez pas de changer ce qui suit en conséquence) et créez un nouveau fichier. Appelez-le info.php. Procédez de la façon suivante :

```
user:/ > cd /web/htdocs/  
user:/web/htdocs > touch info.php
```

Ouvrez le nouveau fichier "info.php" dans un éditeur et ajoutez ce qui suit :

```
<?PHP  
echo phpinfo();  
?>
```

Note: tapez-le exactement de cette manière, y compris les parenthèses !

Enregistrez le fichier et fermez-le. Et maintenant, la partie intéressante ! Ouvrez l'URL suivant dans votre navigateur :

http://127.0.0.1/info.php

ou

http://localhost/info.php

ou

http://rechnername/info.php

ou

http://lokale_IP/info.php

Si à ce stade vous voyez la sortie de phpinfo(), tout s'est déroulé selon les plans et vous pouvez commencer à programmer en PHP. Félicitations ! Vous n'avez plus seulement un système LAMP opérationnel mais aussi un serveur web.

Note: vous pouvez maintenant créer autant de sous-répertoires que vous le souhaitez dans le répertoire /web/htdocs (ou celui que vous avez défini pendant l'installation). Si vous avez par exemple, des projets web multiples, vous pouvez créer un répertoire par projet.

Sachez que /web/htdocs (ou le répertoire spécifié) est le répertoire racine du serveur web. C'est pourquoi l'URL est `http://127.0.0.1/info.php`. Si vous avez d'autres sous-répertoires, par exemple, projet1, projet2, etc, vous devrez adapter l'URL en conséquence : `http://127.0.0.1/project1/` ou `http://127.0.0.1/project2/`, etc.

Note: les fichiers PHP sont analysés (affichés/exécutés) seulement s'ils se trouvent dans ces chemins, c'est-à-dire soit /web/htdocs, soit ses sous-répertoires !

Vous pouvez ajouter ce que vous voulez pour étendre les capacités de votre serveur web.

Recommandations

Dans ce chapitre, je recommande quelques outils d'administration qui faciliteront votre travail sur le système et le serveur web :

Webmin 1.000
(<http://www.webmin.com/>)

Webmin permet de gérer le système très facilement. Vous pouvez aussi l'utiliser pour configurer votre serveur web, par exemple, pour lancer le serveur à chaque démarrage du système. Même chose pour MySQL. Webmin est lui-même simple à utiliser et pour cette raison convient parfaitement aux novices des serveurs web.

Téléchargement :
<http://prdownloads.sourceforge.net/webadmin/webmin-1.000.tar.gz?download>
ou
<http://www.webmin.com/>

phpMyAdmin 2.3.1
(<http://www.phpwizard.net/projects/phpMyAdmin/>)

phpMyAdmin est un très bon outil pour MySQL. Il permet de créer, d'effacer, d'éditer des tables et plus encore. Fortement recommandé.

Téléchargement :
([phpMyAdmin-2.3.1-php.tar.gz](#))
<http://www.phpmyadmin.net/index.php?dl=2>

Configuration par Webmin

Définitions pour le serveur Apache::
webmin => Server => Apache Webserver
Module config:

Apache server root directory:

/usr/local/apache/1.3.27/bin/

Path to httpd executable:

/usr/local/apache/1.3.27/bin/httpd

Apache version:

sélectionnez le champ vide et tapez : => **1.3.27**

Path to apachectl command:

dans le champ vide => **/usr/local/apache/1.3.27/bin/**

Command to start Apache:

dans le champ vide => **/etc/init.d/apachectl start**

Note: si apachectl n'est pas dans le répertoire, copiez-le :

root > cp /usr/local/apache/1.3.27/bin/apachectl /etc/init.d/

Command to stop Apache:

dans le champ vide => **/etc/init.d/apachectl stop**

Display virtual servers as:

=> **Icons**

Order virtual servers by :

=> **order in config file(s)**

Maximum number of servers to display

=> **100**

Path to httpd.conf

dans le champ vide => **/usr/local/apache/1.3.27/conf/httpd.conf**

Path to srm.conf

dans le champ vide => **/usr/local/apache/1.3.27/conf/srm.conf**

Path to access.conf

dans le champ vide => **/usr/local/apache/1.3.27/conf/access.conf**

Path to mime.types

dans le champ vide => **/usr/local/apache/1.3.27/conf/mime.types**

File to add virtual servers to:

=> **httpd.conf**

Test config file before applying changes?

=> **Yes**

Note: ne tapez pas le => !

Si vous voulez lancez automatiquement le serveur Apache au démarrage du système, vous pouvez le définir dans Webmin de cette manière :

webmin => System => Bootup and Shutdown

Si Apache ne figure pas ici, ajoutez-le comme nouveau service.

Name => **apachectl**

Script => **habituellement chargé automatiquement**

Start at boot time? => **Yes**

Bootup commands => **/etc/init.d/apachectl start**

Shutdown commands => **/etc/init.d/apachectl stop**

Mise à jour

Mettre à jour PHP

Copiez d'abord le fichier compressé de la nouvelle version de PHP dans le répertoire suivant :

```
/usr/local/src/lamp/
```

Nous supposons que la nouvelle version de PHP se nomme PHP 4.2.4 (J'ignore s'il y aura un jour une telle version). Cette partie décrit les étapes à suivre pour mettre PHP à jour. En gros, il suffit de permuter l'ancienne et la nouvelle version.

Décompressez le fichier que vous venez de copier comme ceci.

Ouvrez une console texte(un terminal shell, e.g. Bash), et exécutez les commandes:

```
user:~ ➤su
```

```
[Tapez votre mot de passe root]
```

```
root:~ ➤cd /usr/local/src/lamp/
```

```
root:/usr/local/src/lamp > tar -xvzf php-4.2.4.tar.gz
```

Si l'ancien répertoire de source sous "/usr/local/src/lamp" existe toujours, procédez comme suit.

Notre ancien répertoire source se nomme "/usr/local/lamp/php-4.2.3".

Nous devons d'abord faire une copie de libphp4.so. Donnez-lui le nom "libphp4-4.2.3 .so" de cette manière :

```
root:/ > cd /usr/local/apache/1.3.27/libexec/
```

```
root:/usr/local/apache/1.3.27/libexec > cp libphp4.so libphp4-4.2.3.so
```

Nous créons une copie de sauvegarde de l'ancien fichier php.ini file, comme suit :

```
root:/ > cd /  
root:/ > cp /usr/local/lib/php.ini /usr/local/lib/php-4.2.3.ini
```

Nous supprimons l'ancien fichier php.ini puisqu'il est quand même plus sensé d'utiliser le nouveau :

```
root:/ > rm /usr/local/lib/php.ini
```

C'est une excellente idée d'avoir conservé l'ancien répertoire source de PHP puisque ça vous évite une grande quantité de frappe !

Ceci est dû au fait que l'ancien répertoire, "/usr/local/lamp/php-4.2.3", contient un court script Shell. Avant la dernière installation, ce script a stocké tous les paramètres de ./configure. Par conséquent, si vous n'aviez pas conservé l'ancien répertoire, vous seriez obligés de taper tous ces paramètres manuellement !

Exécutez maintenant ./configure, comme suit :

```
root:/ > cp /usr/local/lamp/php-4.2.3/config.nice /usr/local/lamp/php-4.2.4/config.nice  
root:/ > cd /usr/local/lamp/php-4.2.4  
root:/usr/local/lamp/php-4.2.4 > ./config.nice  
root:/usr/local/lamp/php-4.2.4 > make  
root:/usr/local/lamp/php-4.2.4 > make install
```

Note: si vous n'avez pas conservé l'ancien répertoire "/usr/local/lamp/php-4.2.3", vous devez taper tous les paramètres de ./configure, comme indiqué dans la partie "Installation => PHP 4.2.3" ci-dessus.

Copiez maintenant le nouveau php.ini dans le bon répertoire :

```
root:/ > cd /  
root:/ > cp /usr/local/lib/php.ini-dist /usr/local/lib/php.ini
```

Adaptez et modifiez le nouveau fichier php.ini selon vos besoins, comme indiqué dans la partie "Configuration => php.ini" ci-dessus.

Redémarrez Apache et la mise à jour est terminée :

```
root:/ > /usr/local/apache/1.3.27/bin/apachectl restart
```

Commentaires de fin

Postface

Comme nous le savons tous, personne n'est parfait et il y a peut-être des erreurs dans ce tutoriel. Si un expert en la matière lit ce tutoriel et pense que des corrections doivent être apportées, qu'il manque quelque chose ou qu'il faille ajouter des explications supplémentaires, qu'il n'hésite pas à me le dire de façon à améliorer ce tutoriel. Beaucoup de soin et d'attention ont été apportés à la création de ce tutoriel et il a été testé sur plusieurs systèmes SuSE Linux 8.0. Mais il devrait être utilisable pour d'autres distributions Linux.

Il existe certainement de nombreuses autres manières de configurer un système LAMP, mais ce tutoriel est avant tout destiné à aider les débutants dans l'installation d'un serveur web. J'essaierai de tenir à jour ce tutoriel. Je vous suggère de visiter mon site souvent afin de vérifier si une version mise à jour est disponible (voir le commentaire à côté du lien de téléchargement).

Références

- Je maintiendrai ce tutoriel à jour à cette adresse: <http://linux.computerbraxas.de/> [en Allemand]
- <http://www.apache.org/>
- http://www.apache.org/dist/httpd/apache_1.3.27.tar.gz
- <http://www.mysql.org/>
- <http://www.gzip.org/zlib/>
- GD: <http://www.boutell.com/gd/>
- <http://www.pdfliib.com/pdfliib/index.html>
- <http://www.php.net/>
- <http://www.webmin.com/>

Site Web maintenu par l'équipe d'édition LinuxFocus © Sascha Blum "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Translation information: de --> -- : Sascha Blum (homepage) de --> en: Orla Shanaghy < o_shanaghy(at)yahoo.com > en --> fr: Georges Tarbouriech < gt(at)linuxfocus.org >
---	---