LinuxFocus article number 270
http://linuxfocus.org

by Sascha Blum (homepage)

About the author:
I really like using Linux because it?s extremely powerful and stable, and especially because it gives the user so many options and resources. Best of all, it's open to everyone (OpenSource) and so everyone can get involved in developing it.

Translated to English by:
Orla Shanaghy
<o_shanaghy(at)yahoo.com>

Installing a LAMP System

Abstract:

In this tutorial, I would like to show you how to install a Linux server with basically every useful feature included. In other words, I will show you how to install a LAMP system.

But first I'll tell you what the abbreviation LAMP stands for.
LAMP means Linux Apache MySQL PHP. So, as you might guess from the name, a LAMP system consists of a Linux operating system, an Apache Web server, a MySQL database, and the script language PHP.

Introduction

This tutorial explains how to install a LAMP system using Dynamic Shared Objects (DSO).

DSOs have a major advantage over static installation: you can replace each individual module with a newer version easily and at any time, without having to recompile and reinstall all the other modules. It doesn't matter whether the module in question is the PDF-Lib module, the GD-Lib module, or anything else. With a static installation, if you wanted to update PHP 4.2.3 to PHP 4.2.4, for example, you would have to recompile and reinstall everything - and by this I mean the Apache server, the GD-Lib, the PDF-Lib, and all the other modules you need (and of course PHP itself). With a DSO installation, only PHP would be affected, and everything else remains the same.

Note: in general, you should carefully read the README file for each package before installing or

compiling, as every installation can be different. Often, a successful installation depends on some switch or other that you have to or can set using ./configure. Having said that, based on my testing, this installation should work first time round. If you get any errors, consult the README. Make sure to use the root access permissions for the installation!
But enough preamble. Let's get started with installing our LAMP system.

Make sure to read this tutorial carefully and in its entirety before starting the installation!

What You Need and Download Sources

You need the following packages, which you should download before starting the installation:

- Apache 1.3.27
  (http://www.apache.org/)
  Direct download:
  http://www.apache.org/dist/httpd/apache_1.3.27.tar.gz (2,2 MB)

- MySQL
  (http://www.mysql.org/)
  RedHat packages (rpm):
  MySQL 3.23.52 Server (i386) (7.4M)
  MySQL 3.23.52 Client programs (i386) (2.2M)
  MySQL 3.23.52 Libraries and Header files for development (i386) (743K)
  MySQL 3.23.52 Client shared libraries (i386) (232K)

- zlib 1.1.4
  (http://www.gzip.org/zlib/)
  Download:
  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/)
  Download:
  http://www.boutell.com/gd/http/gd-1.8.4.tar.gz (252 KB)
  Note: for licensing reasons, the GD library no longer supports the GIF format (and has not done so for quite some time)!

- PDF Lib 4.0.3
  (http://www.pdflib.com/pdflib/index.html)
  Download:
  http://www.pdflib.com/pdflib/download/pdflib-4.0.3-Linux.tar.gz (3,2 MB)

- PHP 4.2.3
  (http://www.php.net/)
  Download:
  http://us3.php.net/do_download.php?download_file=php-4.2.3.tar.gz (3,3 MB)

Installation

Once you have downloaded all these packages, you're ready to go. First, copy the files to the following directory (if you have not already done so, create the directory lamp using mkdir/usr/local/src/lamp):

/usr/local/src/lamp/

The only files you don't need to copy here are the MySQL RPM files. They can be installed straight away in the usual way. The best idea is to do this first. For instructions on how to do this, see the section "MySQL 3.23.52" below.

Now all six packages should be in the /usr/local/src/lamp/ directory as tar.gz. Now you need to unpack them. Proceed as shown below.

Note: the commands you need to enter appear in bold type; PC output is in normal type. All input is preceded by a > symbol.

First open a text console (shell terminal, e.g. Bash), then execute the following commands:

user:~ >**su**
[Now enter your root password]
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**

After you have unpacked all the packages, enter the command "**ls -l**" to display all the directories.

From this point on, it is essential that you follow the installation steps exactly in the order shown here. This is because some packages need other packages to work properly. For example, the GD library needs zlib and libpng, and libpng in turn needs zlib. Now let's move on to the Apache Web server.

Apache 1.3.27

Note: make sure to read the README file! There are several switches under ./configure that can be set here.

Never compile the Apache Web server using the option --enable-module=all! If you do it this way, nothing will work. The best way to go about it is to specify as few modules as possible. Usually, this is more than enough for DSO support. You can then add any other modules you want yourself, which is after all the advantage of the DSO installation.

To install and configure Apache, proceed as follows.

First, change to a text console (shell terminal, e.g. Bash), as before.

Note: do not enter user:/usr/local/src/lamp > with your commands. This is the Linux prompt and is Linux's way of telling you that it is waiting for input. Your prompt may look different, as it can be individually configured.

user:/usr/local/src/lamp > **cd apache_1.3.27**
user:/usr/local/src/lamp/apache_1.3.27 > **su**
[Enter your root password]
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: enter this last part as one line! There is usually a space character in front of the --. The document directory where your websites will be stored later comes after -datadir. You can of course choose your own document directory. If you enter a document directory other than "/web/htdocs", though, make sure to change the relevant paths accordingly later in this tutorial.

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

If you have entered everything correctly, your Apache Web server should now be completely compiled and installed.

MySQL 3.23.52

If you followed the instruction earlier in this tutorial, this has already been installed.

Security note: if your server is connected to a public network, i.e. an intranet or the internet, make sure to make the password for the MySQL Server root user as complicated as possible!

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: replace /home/user/donwload/mysql with the directory where the relevant RPM files are located.

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**

Comment: we use the switch --shared here to tell zlib that we want to include the library as a dynamic module in PHP.

libpng 1.2.3

The installation for libpng is a little different from the usual. First, change to the directory /libpng-1.2.3/scripts/ :

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

Then enter the following commands:

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 ..**

With these commands, you have just copied the relevant make file into the libpng master directory. Now you need to take a look at the make file and make any changes that the system may require, e.g. special include directories. Normally, all the data in the file is correct, but you should still check, as this allows you to find errors more quickly.

To continue, enter the following command:

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

If you do not get any error messages at this point, you can now install libpng with the following command:

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

gd-1.8.4

First, change into the directory gd-1.8.4:

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

You should also take a look at the make file here. If something in your system has changed, you will have to make the corresponding changes to the make file now. You can view and edit the file using any text editor you like.

Usually, though, you do not need to make any changes to the make file.

If you are now happy with the make file, enter the following command:

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

If any errors occur at this point, enter the following:

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

But only enter this last command if there are errors! If you execute make clean, you will have to check the make file again and adapt it accordingly, then carry out the make again.

Note: make sure to check the settings for INCLUDEDIRS and LIBDIRS!

PDF-Lib 4.0.3

This is a little simpler, as the module is already compiled and you only have to copy it to the directory /usr/local/lib.

To do this, enter the following:

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

Lastly, you have to install PHP.

Change into the PHP directory:

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/lo cal/lib**

--with-zlib-dir=/usr/local/lib --with-mysql --with-xml

Note: enter this last part as one line! There is usually a space character in front of the --. There is not an error in the second and third lines ("sysvshm" and "sysvsem").

Then enter the following, as before:

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

Note: compiling (make) PHP can take a little longer on slow PC systems. So don't get impatient if nothing appears to be happening for long periods. You can delete the directory /usr/local/src/lamp (as root) using "rm -r /usr/local/src/lamp". Make sure to enter this correctly, because if you execute a "rm -r /" as root, you will destroy the whole system. However, be aware that if you delete "/usr/local/src/lamp", it will be more work to deinstall or update the system. Therefore, you should only delete the packed source package ".tar.gz" and retain the directories with the sources.

Configuration

httpd.conf

So, that was the installation. Now for the configuration.

First of all, we have to tell the Apache Web server what it is supposed to do with the *.php- or *.php3 files.

To do this, change into the Apache "conf" directory:

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 >

Then, open the "httpd.conf" file in a text editor so you can edit and then save it.

Note: the editor "Kate" is very suitable for editing the config file. Note that KDE has to be running in the background. To start it, press Alt + F2 => kdesu kate. Press Ctrl + G to go to the line you want.

In the file, you will find the following around line 190:

#
#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.

At this point, enter the following, if it is not there already:

LoadModule php4_module libexec/libphp4.so

You will find the following around line 770:

#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

At this point, add the following:

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: make sure to enter this accurately, or errors may occur.

If you do not want the PHP parser to run HTML files, you can omit the following lines:

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

Now the httpd.conf file is configured.

php.ini

Now you have to set up, and possibly adapt, the php.ini file.

First, you have to copy the php.ini file to the proper location. To

What is a parser?

A parser is simply a piece of software that interprets text. The text in question can be source

First, you have to copy the php.ini file to the proper location. To do this, change into the PHP install directory:

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

Now copy the file "php.ini-dist" into the directory /usr/local/lib and re-name the file "php.ini". Do this as follows:

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

Then write "pdflib" into the php.ini file as an extension. This is so that PHP knows what to do with the corresponding PDF functions, should you ever need these and want to work with them. PHP finds the other modules on its own (zlib, GD, etc.).

Now open the file "/usr/local/lib/php.ini" in a text editor. The section about extensions is located around line 371.

It should look something like this:

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

extension_dir = ./ <= remove this and replace it with the following:

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

Now save the file.

You're finished - you now have a complete, fully-functioning LAMP system!

Now for the server test. This tests whether you can start the server successfully. The first step is to shut down any old servers that might still be running (if a Web server was already installed when you installed the distribution, for example). To do this, enter the following:

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

Now attempt to start the new server, as follows:

code (like C++) or a document markup language (like HTML). The parser checks the text for syntactic and semantic errors, and passes on the parsed text, usually in an efficient and compact internal code, to the processing application.

Text that is run through a parser takes a little longer to display. This means that pure HTML pages are loaded and displayed quicker than PHP pages or scripts. However, the user does not notice much delay. A delay only becomes noticeable if several users are accessing the same thing, e.g. if several users call up a PHP page or script at the same time, it can take longer to display the page or script, depending on the hardware. Therefore, if you intend to make your LAMP system publicly available, e.g. to connect it to the internet, an intranet, or a network, you should get yourself a powerful, fast computer, otherwise the system may get pretty slow. If, on the other hand, you want to use your LAMP system to develop PHP pages or scripts in conjunction with a MySQL database, you can safely do this using an older PC or notebook. The same applies if you are the only one executing or displaying PHP pages or scripts on your computer system.

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

If you see the following message...

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

... everything is OK and your server is up and running!

Now change into your "web/htdocs" directory (DocumentRoot - if you have given this a different name, remember to change the following accordingly) and create a new file there. Call the new file info.php. To do this, proceed as follows:

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

Open the new file "info.php" in an editor and write in the following:

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

Note: make sure to enter this exactly as it appears here, including the brackets!

Save the file and close it. Now for the exciting part! Open the following URL in your Internet browser:

http://127.0.0.1/info.php
or
http://localhost/info.php
or
http://computer_name/info.php
or
http://local_IP_address/info.php

At this point, if you can see the output of phpinfo(), everything has gone according to plan and you can get on with programming in PHP straight away. Congratulations! You now not only have a fully-functioning LAMP system but also a Web server to boot.

Note: you can now create as many sub-directories as you like in the directory /web/htdocs (or any other directory you specified during installation). For example, if you have multiple Web projects, you can create a directory for each project.

Note that /web/htdocs (or the other directory you specified) is your root directory as far as the Web server is concerned. This is why the URL is http://127.0.0.1/info.php. If you have other sub-directories,

e.g. project1, project2, etc., you will have to adjust the URL accordingly: http://127.0.0.1/project1/ or http://127.0.0.1/project2/, etc.

Note: PHP files are parsed (displayed/executed) only if they are located in these paths, i.e. either /web/htdocs or its subdirectories!

You can add to and extend your Web server in any way you like.

Recommendations

In this section I recommend some admin tools that will make your work with the system and the Web server significantly easier:

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

Webmin lets you handle your system with absolute ease. You can also use it as an easy way to configure your Web server, e.g. to have your server restart with every system start. The same goes for MySQL. Webmin itself is simple to use and for this reason is ideally suited to Web server novices.

Download:
http://prdownloads.sourceforge.net/webadmin/webmin-1.000.tar.gz?download
or
http://www.webmin.com/

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

phpMyAdmin is a great tool for MySQL. It lets you create, delete, and edit tables, and a lot more. Also highly recommended.

Download:
(phpMyAdmin-2.3.1-php.tar.gz)
http://www.phpmyadmin.net/index.php?dl=2

Configuration using Webmin

Settings for the Apache server::
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:
select empty field and enter: => **1.3.27**

Path to apachectl command:
in empty field => **/usr/local/apache/1.3.27/bin/**

Command to start Apache:
in empty field => **/etc/init.d/apachectl start**

Note: if apachectl is not in the directory, just copy it there:
root > cp /usr/local/apache/1.3.27/bin/apachectl /etc/init.d/

Command to stop Apache:
in empty field => **/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
in empty field => **/usr/local/apache/1.3.27/conf/httpd.conf**

Path to srm.conf
in empty field => **/usr/local/apache/1.3.27/conf/srm.conf**

Path to access.conf
in empty field => **/usr/local/apache/1.3.27/conf/access.conf**

Path to mime.types
in empty field => **/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: do not enter the => !

If you want the Apache server to start automatically when the system boots up, you can set this up in Webmin as follows:
webmin => System => Bootup and Shutdown

If Apache is not listed here, just enter it as a new service.

Name => **apachectl**
Script => **is usually loaded automatically**
Start at boot time? => **Yes**
Bootup commands => **/etc/init.d/apachectl start**
Shutdown commands => **/etc/init.d/apachectl stop**

Update

Updating PHP

First of all, copy the packed file of the new PHP version into the following directory:

/usr/local/src/lamp/

Let’s assume that the new PHP version is called PHP 4.2.4 (I don’t know whether there will ever actually be a version of this name). This section describes the steps you need to take to update PHP. Basically, all you need to do is swap the old version for the new version.

Unpack the file you just copied, as follows.

Open a text console (shell terminal, e.g. Bash), and execute the following commands:

user:~ >**su**
[Only enter your root password]
root:~ >**cd /usr/local/src/lamp/**
root:/usr/local/src/lamp > **tar -xvzf php-4.2.4.tar.gz**

If your old source directory under "/usr/local/src/lamp" still exists, proceed as follows.

Our old source directory is called "/usr/local/lamp/php-4.2.3".

We first need to create a copy of libphp4.so. Give the copy the name "libphp4-4.2.3 .so". Do this as follows:

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**

We then create a backup of the old php.ini file, as follows:

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

Then delete the old php.ini file, as it makes more sense to use the new one:

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

It’s a very good thing that you kept your old PHP source directory, as you have saved yourself a lot of typing!

This is because the old directory, "/usr/local/lamp/php-4.2.3", contains a short Shell script. Before the last installation, this script stored all the parameters from ./configure. Therefore, if you had not kept this old directory, you would now have to enter all these parameters by hand!

Now execute ./configure, as follows:

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: also, if you did not keep the old directory "/usr/local/lamp/php-4.2.3", you will have to enter all the ./configure parameters, as described in the section "Installation => PHP 4.2.3" above, by hand.

Now copy the new php.ini into the correct directory:

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

Now just adapt and change the new php.ini file as necessary, as described in the section "Configuration => php.ini" above.

Finally, restart Apache, and that’s your update completed:

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

Closing Comments

Postscript

As we all know, no-one is perfect, and there may be errors in this tutorial. If a subject matter expert has read this tutorial and thinks that anything needs to be corrected, that something is missing, or needs further explanation, please let me know, so that I can improve the tutorial. A lot of care and attention went into the creation of this tutorial and it was successfully tested on several systems with SuSE Linux 8.0. But you should also be able to use it with other Linux distributions.

There are certainly a lot more ways to configure a LAMP system than explained here, but this tutorial is basically intended as an aid for beginners in setting up a Web server. I will try to keep the tutorial up to date. I suggest you take a look at my homepage every so often to check whether an updated version is available (see the comment beside the download link).

References

- I will keep this tutorial up to date here: http://linux.computerbraxas.de/ [in German]
- 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.pdflib.com/pdflib/index.html
- http://www.php.net/
- http://www.webmin.com/

2005-01-14, generated by lfparser_pdf version 2.51