

# 用源代码搭建 **LAMP** 环境

By Mark

[kevin.zwf@gmail.com](mailto:kevin.zwf@gmail.com)

## 版权声明：

本文遵循“署名-非商业性使用-相同方式共享 2.5 中国大陆”协议

您可以自由复制、发行、展览、表演、放映、广播或通过信息网络传播本作品

您可以根据本作品演绎自己的作品

您必须按照作者或者许可人指定的方式对作品进行署名。

您不得将本作品用于商业目的。

如果您改变、转换本作品或者以本作品为基础进行创作，您只能采用与本协议相同的许可协议发布基于本作品的演绎作品。

对任何再使用或者发行，您都必须向他人清楚地展示本作品使用的许可协议条款。

如果得到著作权人的许可，您可以不受任何这些条件的限制。

Mark Zhu ([kevin.zwf@gmail.com](mailto:kevin.zwf@gmail.com))

## 本文目录

第 1 章 用源代码搭建 LAMP 环境.....	2
第 1.1 节 项目简介.....	2
第 1.2 节 前期准备.....	2
第 1.3 节 安装 mysql.....	5
第 1.4 节 编译 apache.....	8
第 1.5 节 编译 php.....	9
第 1.6 节 安装 php 扩展模块工具 .....	11
第 1.7 节 配置 apache 与 php 集成.....	13
第 1.8 节 测试 PHP 并提高安全性 .....	15
第 1.9 节 登录 xcache 管理页面查看 xcache 状态.....	15
第 1.10 节 安装、配置 Discuz!.....	16

---

## 第1章 用源代码搭建 LAMP 环境

### 第 1.1 节 项目简介

LAMP 是一个缩写,它指一组通常一起使用来运行动态网站或者服务器的开源软件,包括:Linux 操作系统,APACHE 服务器,MYSQL,Perl,PHP 或者 Python 编程语言。

虽然这些开放源代码程序本身并不是专门设计成同另外几个程序一起工作的,但由于它们都是影响较大的开源软件,拥有很多共同特点,这就导致了这些组件经常在一起使用在过去的几年里,这些组件的兼容性不断完善,在一起的应用情形变得更加普遍。并且它们为了改善不同组件之间的协作,已经创建了某些扩展功能。目前,几乎在所有的 Linux 发行版中都默认包含了这些产品。Linux 操作系统、Apache 服务器、MySQL 数据库和 Perl、PHP 或者 Python 语言,这些产品共同组成了一个强大的 Web 应用程序平台。

本章的目标是完全通过源代码编译安装,组建一个 LAMP 的环境,并且在上面运行一个使用 PHP 语言编写的 Discuz 论坛。

### 第 1.2 节 前期准备

如果你在一台机器启动两个 apache 和 mysql,那么很可能造成一定的冲突,所以为了减少不必要的麻烦,我这边首先停止卸载他们。

由于整个环境会涉及比较多的依赖关系包,所以我们将一些依赖包装上。(根据不同的环境可能不仅限于这些包)

```
gcc
gcc-c++
flex
bison
autoconf
automake
libpng-devel
libjpeg-devel
freetype-devel
openjpeg-libs
fontconfig-devel
```

```
libXpm-devel  
cmake  
ncurses-devel  
apr-devel  
pcre-devel  
apr-util-devel  
openssl-devel  
libxml2-devel  
curl-devel
```

除了这些 redhat 光盘自带的包以外，还有一些其他的源码包。

#### 1 : GD2

GD2 是一个生成图片所使用的依赖包

```
yum localinstall -y gd-devel-2.0.35-26.el7.x86_64.rpm
```

#### 2 : libmcrypt 与 mhash 是 php 会依赖的一些包

```
yum localinstall -y libmcrypt-2.5.8-13.el7.x86_64.rpm libmcrypt-devel-2.5.8-  
13.el7.x86_64.rpm mhash-0.9.9.9-10.el7.x86_64.rpm mhash-devel-0.9.9.9-  
10.el7.x86_64.rpm
```

### 第 1.3 节 安装 *mysql*

Mysql 这些选择的是 mysql5.6 版本，此版本一改原有的编译方式，使用 cmake 代替了原来的 configure。

```
[root@server mysql]# groupadd -g 27 mysql
```

```
[root@server mysql]# mkdir /var/lib/mysql
[root@server mysql]# useradd mysql -u 27 -g mysql -s /sbin/nologin -d /var/lib/mysql/
[root@server mysql]# chown mysql. /var/lib/mysql/
[root@server mysql]# chmod 700 /var/lib/mysql/
[root@server mysql]# tar xf mysql-5.6.26.tar.gz
[root@server mysql]# cd mysql-5.6.26/
[root@server mysql]# cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql \
    -DMYSQL_UNIX_ADDR=/var/lib/mysql/mysql.sock \
    -DDEFAULT_CHARSET=utf8 \
    -DDEFAULT_COLLATION=utf8_general_ci \
    -DWITH_EXTRA_CHARSETS=all \
    -DWITH_MYISAM_STORAGE_ENGINE=1 \
    -DWITH_INNOBASE_STORAGE_ENGINE=1 \
    -DMYSQL_DATADIR=/var/lib/mysql
[root@server mysql]# make
[root@server mysql]# make install
[root@server mysql]# cp /usr/local/mysql/support-files/my-default.cnf /etc/my.cnf
[root@server mysql]# echo 'export PATH="/usr/local/mysql/bin:$PATH" ' >> ~/.bashrc
[root@server mysql]# . ~/.bashrc
[root@server mysql]# /usr/local/mysql/scripts/mysql_install_db --user=mysql
--datadir=/var/lib/mysql --basedir=/usr/local/mysql --builddir=/usr/local/mysql
[root@server mysql]# cp support-files/mysql.server /etc/init.d/mysqld
[root@server mysql]# chmod 755 /etc/init.d/mysqld
[root@server mysql]# service mysqld start
[root@server mysql]# mysqladmin password 'uplooking'
```

常见 cmake 参数介绍

cmake . -LH

查看帮助，类似于./configure -help

-DCMAKE\_INSTALL\_PREFIX=

指定安装主目录

-DMYSQL\_DATADIR

指定库文件存放目录

-DMYSQL\_UNIX\_ADDR

指定 socket 文件所在路径

-DDEFAULT\_CHARSET

指定使用的默认字符编码

-DDEFAULT\_COLLATION

指定使用的默认字符规则

-DWITH\_EXTRA\_CHARSETS

其他字符码支持

-DWITH\_MYISAM\_STORAGE\_ENGINE

编译 myisam 存储引擎

-DWITH\_INNOBASE\_STORAGE\_ENGINE

编译 innodb 存储引擎

-DWITH\_READLINE

增加 readline 支持，通过 readline，可以方便的在命令行上面移动，增删，复制，粘贴，搜索。



## 第1.4节 编译 *apache*

```
tar xf httpd-2.4.16.tar.gz
cd httpd-2.4.16
./configure --prefix=/usr/local/apache2 \
    --enable-cache \
    --enable-cache-disk \
    --enable-so \
    --enable-sed \
    --enable-http \
    --enable-mpms-shared=all \
    --enable-cgid \
    --enable-rewrite \
    --enable-vhost-alias \
    --enable-ssl
make
make install
/usr/local/apache2/bin/apachectl start
```

访问 <http://localhost> 测试

## 第1.5节 编译 *php*

```
tar xf php-5.6.13.tar.gz
cd php-5.6.13
./configure --prefix=/usr/local/php \
    --with-config-file-path=/usr/local/php/etc \
    --with-mysql=/usr/local/mysql \
    --with-config-file-scan-dir=/usr/local/php/etc/php.d \
```

```
--with-mysql=/usr/local/mysql/bin/mysql_config \  
--with-freetype-dir \  
--with-jpeg-dir \  
--with-png-dir \  
--with-zlib \  
--with-libxml-dir \  
--enable-xml \  
--with-pdo-mysql=/usr/local/mysql/ \  
--enable-bcmath \  
--enable-shmop \  
--enable-sysvsem \  
--enable-inline-optimization \  
--with-curl \  
--enable-mbregex \  
--enable-mbstring \  
--with-gd \  
--enable-gd-native-ttf \  
--with-openssl \  
--enable-pcntl \  
--enable-sockets \  
--enable-zip \  
--enable-soap \  
--with-apxs2=/usr/local/apache2/bin/apxs  
make  
make install  
mkdir /usr/local/php/etc  
cp php.ini-production /usr/local/php/etc/php.ini
```

php 常见参数介绍

--with-config-file-path 与 --with-config-file-scan-dir 在指定 php 配置文件的路径

--with-mysql 和 --with-mysqli 在指定你的 mysql 的位置和它的相关工具

--with-iconv-dir, --with-freetype-dir, --with-jpeg-dir, --with-png-dir, --with-gd, --with-zlib, --with-libxml-dir 这些都是在启用对某种文件的支持

--with-curl 和 --with-curlwrappers 是用于支持 curl 函数，此函数允许你用不同的协议连接和沟通不同的服务器

--with-openssl, --with-mhash, --with-mcrypt 这都是和加密有关的参数，启用它们是为了让 php 可以更好的支持各种加密。

"--enable-bcmath" 高精度数学运算组件。

"--enable-shmop" 和 "--enable-sysvsem" 使得你的 PHP 系统可以处理相关的 IPC 函数。IPC 是一个 Unix 标准通讯机制，它提供了使得在同一台主机不同进程之间可以互相通讯的方法。

"--enable-inline-optimization" 栈堆指针和优化线程。

"--enable-pcntl" 多线程优化。

### 第1.6节 安装 *php* 扩展模块工具

XCache 是一个开源的 opcode 和 var 的缓存器，其工作原理是通过内存来缓存 php 的编译结果，当用户读取 php 网页时后台的 php 编译器不参加编译工作，而是将内存中的结果直接供用户读取，从而减轻频繁编译对服务器造成的压力。与同类的 apc、eaccelerator 相比 XCache 具有覆盖速度快、大流量/高负载的机器上运行稳定、能够快速跟进 php 版本。

安装 xcache 之前先安装 xcache 的一个依赖包 re2c。

```
tar xf re2c-0.13.5.tar.gz
cd re2c-0.13.5
./configure
make
make install
```

之后安装 xcache

```
tar xf xcache-3.2.0.tar.gz
cd xcache-3.2.0/
```

```
/usr/local/php/bin/phpize
./configure --enable-xcache \
    --with-php-config=/usr/local/php/bin/php-config \
    --enable-xcache-optimizer \
    --enable-xcache-assembler \
    --enable-xcache-encoder \
    --enable-xcache-decoder
make && make install
cp -r htdocs/ /usr/local/apache2/htdocs/xcache
chmod 755 /usr/local/apache2/htdocs/xcache -R

mkdir /usr/local/php/etc/php.d
cp xcache.ini /usr/local/php/etc/php.d/
vi /usr/local/php/etc/php.d/xcache.ini
```

找到行内容

```
extension = xcache.so
```

修改为

```
extension = /usr/local/php/lib/php/extensions/no-debug-zts-20131226/xcache.so
```

访问 <http://<yourdomain>/xcache/cacher/mkpassword.php>  
生成密码

找到行

```
xcache.admin.user = "mOo"
xcache.admin.pass = ""
```

修改为

```
xcache.admin.user = "mark"
xcache.admin.pass = "026d9f9617a5e636b9f586eba5adbda6"
```

xcache.admin.pass 是一个通过 md5 加密之后的密码，范例中为'uplooking'。

### 第 1.7 节 配置 *apache* 与 *php* 集成

修改 apache 主配置文件

```
vi /usr/local/apache2/conf/httpd.conf
```

找到:

```
AddType application/x-gzip .gz .tgz
```

在该行下面添加

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

找到:

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

修改为

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

找到:

```
#Include conf/extra/httpd-mpm.conf
#Include conf/extra/httpd-info.conf
#Include conf/extra/httpd-vhosts.conf (如果需要做虚拟主机，打开这行参数)
#Include conf/extra/httpd-default.conf
```

去掉前面的“#”号,取消注释。

注意:以上 4 个扩展配置文件中的设置需要根据实际需要进行更改。

```
/usr/local/apache2/bin/apachectl stop
/usr/local/apache2/bin/apachectl start
```

## 第 1.8 节 测试 *PHP* 并提高安全性

在网站根目录放置 phpinfo.php 脚本,检查 phpinfo 中的各项信息是否正确。

```
#vi phpinfo.php
<?php
phpinfo();
?>;
```

确认 PHP 能够正常工作后,在 php.ini 中进行设置提升 PHP 安全性。

```
[root @ server1 src] # vi /usr/local/php/etc/php.ini
```

找到:

```
disable_functions =
```

设置为:

```
phpinfo,passthru,exec,system,chroot,scandir,chgrp,chown,escapeshellcmd,escapeshell
arg,shell_exec,proc_open,proc_get_status,error_log,ini_alter,ini_restore,dl,pfsock
open,openlog,syslog,readlink,symlink,leak,popen,passthru,stream_socket_server
```

```
/usr/local/apache2/bin/apachectl stop
/usr/local/apache2/bin/apachectl start
```

## 第 1.9 节 登录 **xcache** 管理页面查看 **xcache** 状态

<http://localhost/admin/index.php>

登录用户名为 mark，密码为 uplooking。如需修改，可以通过

```
vi /usr/local/php/etc/php.d/xcache.ini  
  
xcache.admin.user = "mark"  
  
xcache.admin.pass = "026d9f9617a5e636b9f586eba5adbda6"
```

如果要获取 md5 加密后的密码值可以通过访问

<http://localhost/admin/mkpassword.php>

来生成加密后的密码

## 第 1.10 节 安装、配置 **Discuz!**

从服务器上下载 discuz，自行通过帮助安装配置 discuz。