

## Linux 下 apache+mysql+php 安装配置详细步骤

版本历史

|                            |           |      |     |      |
|----------------------------|-----------|------|-----|------|
| 1. 文档属性                    |           |      |     |      |
| 文件状态                       |           | 文档编号 |     |      |
| [√] 初稿<br>[ ] 发布<br>[ ] 修订 | 版本号       |      | 1.0 |      |
|                            | 文档密级      |      | 公开  |      |
|                            | 采纳标准      |      |     |      |
| 2. 版本历史                    |           |      |     |      |
| 序号                         | 日期        | 版本   | 作者  | 修订说明 |
| 1                          | 2012.5.18 | 1.0  |     |      |
| 2                          |           |      |     |      |
|                            |           |      |     |      |
|                            |           |      |     |      |
|                            |           |      |     |      |

|   |           |
|---|-----------|
| <b>第一章 准备工作 .....</b>                                     | <b>1</b>  |
| 1.1 将所有源代码包下载并复制到一统一目录下，如/USR/LOCAL/SRC/下 .....           | 1         |
| 1.2 检查当前系统是否具有编译工具.....                                   | 1         |
| 1.3 检查系统是否已经默认安装了 APACHE、MYSQL 和 PHP 并卸载.....             | 1         |
| 1.4 在 LINUX 系统中源代码包安装过程 .....                             | 1         |
| 1.5 各个软件编译安装顺序 APACHE、MYSQL、PHP 或者 MYSQL、APACHE、PHP ..... | 2         |
| <b>第二章 MYSQL 安装 .....</b>                                 | <b>3</b>  |
| 2.1 MySQL 安装 .....  | 3         |
| 2.2 配置 MySQL 数据库.....                                     | 3         |
| <b>第三章 APACHE 安装 .....</b>                                | <b>5</b>  |
| 3.1 APACHE 安装 .....                                       | 5         |
| 3.2 测试 APACHE 服务器.....                                    | 5         |
| <b>第四章 PHP 安装 .....</b>                                   | <b>7</b>  |
| 4.1 PHP 库文件安装（可省略） .....                                  | 7         |
| 4.2 PHP 安装 .....  | 9         |
| 4.3 LAMP 环境整合 .....                                       | 10        |
| <b>第五章 总结 .....</b>                                       | <b>12</b> |
| 5.1 APACHE、MYSQL、PHP 配置文件 .....                           | 12        |
| 5.2 配置文件常见参数修改 .....                                      | 12        |
| 5.3 安装常见问题解决 .....  | 12        |
| 5.4 补充安装 .....  | 13        |
| 5.5 ./CONFIGURE -HELP .....                               | 14        |

## 第一章 准备工作

### 1.1 将所有源代码包下载并复制到一统一目录下，如/usr/local/src/下

### 1.2 检查当前系统是否具有编译工具

```
#gcc -v          //出现 gcc 安装配置信息
或者#rpm -qa|grep gcc //出现 gcc 安装包信息
#rpm -q make      //出现：make-3.81-3.el5
#rpm -q gcc        //出现：gcc-4.1.2-48.el5
#rpm -q gcc-c++    //出现：gcc-c++-4.1.2-48.el5
```

如果没有安装编译工具则在解压后的源代码包执行./configure 命令时会报错：

```
error: not acceptable C compiler found in $PATH
```

### 1.3 检查系统是否已经默认安装了 apache、mysql 和 php 并卸载

要卸载的话首先查看服务是否开启：

```
# netstat -tnl
```

80 端口和 3306 端口是否开启

先停服务再卸载，卸载时：

如果是用 rpm 方式安装的，则先查找已经安装的包：

```
#rpm -qa|grep httpd
```

```
#rpm -qa|grep mysql
```

```
#rpm -qa|grep php
```

```
#rpm -e 包名      //卸载
```

```
#rpm -e 包名 --nodeps //考虑到包与包之间的依赖关系，有时要强制卸载
```

这样卸载先后顺序就不用考虑了

如果是使用源代码包安装方式，则停掉服务后直接删除安装目录。我们可以查看各个软件的安装路径：

```
#whereis mysql
```

```
#whereis httpd
```

```
#whereis php
```

然后用删除命令删除#rm -rf 文件目录名

### 1.4 在 Linux 系统中源代码包安装过程

apache、mysql 和 php 的安装都需要三个步骤：

①配置(configure)一是检查系统的环境是否适合安装，二是指定安装哪个模块，在编译阶段就编译哪个模块，不指定的模块就不编译安装。执行每个软件的./configure --help 命令能够查看都有哪些配置选项。在此阶段除了指定要安装的模块，还指定安装目录

②编译(make)

③安装(make install)

## 1.5 各个软件编译安装顺序 apache、mysql、php 或者 mysql、apache、php

注意 php 的安装要放到最后。在安装 php 之前需要安装 php5 需要的最新版本库文件，如 libxml2、libmacypt 以及 GD2 库等文件。安装 GD2 库是为了让 php5 支持 gif、png 和 jpeg 图片格式，所以在安装 GD2 库之前还要先安装最新的 zlib、libpng、freetype 和 jpegsrc 等库文件。

## 第二章 MySQL 安装

### 2.1 MySQL 安装

#### ①解压并进入

```
#cd /usr/local/src
#tar zxvf mysql-5.0.41.tar.gz
#cd mysql-5.0.41
```

#### ②#./configure \ cmake .

```
--prefix=/usr/local/mysql \
--with-extra-charsets=all //安装所有的字符集
```

#### ③#make

#### ④#make install

### 2.2 配置 MySQL 数据库

#### ①创建 MySQL 数据库服务器的配置文件

```
#cd /usr/local/src/mysql-5.0.41
#cp support-files/my-medium.cnf /etc/my.cnf
```

#### ②添加一个 mysql 标准组(这里出于安全的考虑, 推荐用特定用户运行 mysql)

```
#groupadd mysql
```

添加 mysql 用户并加到 mysql 组中

```
#useradd -g mysql mysql
```

#### ③创建数据文件目录并将所有权交给 mysql

```
#/usr/local/mysql/bin/mysql_install_db --user=mysql //执行完后生成 var 目录, 用来存放数据库
```

#### ④启动 mysql 前修改文件权限

```
#cd /usr/local/mysql
#chown -R root.root . (有符号点) #将/usr/local/mysql 所有者/组改为 root
#chown -R mysql.mysql var #将/usr/local/mysql/var 所有者/组改为 mysql
查看改后结果#ll
```

```
drwxr-xr-x 2 root root 4096 05-19 17:57 bin
drwxr-xr-x 3 root root 4096 05-19 17:57 include
drwxr-xr-x 2 root root 4096 05-19 17:57 info
drwxr-xr-x 3 root root 4096 05-19 17:57 lib
drwxr-xr-x 2 root root 4096 05-19 17:57 libexec
drwxr-xr-x 4 root root 4096 05-19 17:57 man
drwxr-xr-x 8 root root 4096 05-19 17:57 mysql-test
drwxr-xr-x 3 root root 4096 05-19 17:57 share
drwxr-xr-x 5 root root 4096 05-19 17:57 sql-bench
```

drwx----- 4 mysql mysql 4096 05-19 18:29 var

⑤启动 mysql

```
#/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

查看 3306 开启情况

```
netstat -tnl
```

或者查看进程

```
#ps -le|grep mysqld
```

⑥查看所有 mysql 参数

```
#/usr/local/mysql/bin/mysqladmin version //显示信息代表成功
```

⑦修改 root 密码

没有密码可以直接登录本机服务器

```
#/usr/local/mysql/bin/mysql
```

```
#mysql>select * from mysql.user\G; //查看默认创建的用户
```

```
#mysql>delete from mysql.user where host!='localhost'; //删除非 localhost 用户
```

刷新授权表

```
#mysql>flush privileges;
```

为 root 添加密码:

```
#mysql>set password for 'root'@'localhost'=password('123456');
```

```
#mysql>select * from mysql.user \G //查看密码是否已经添加
```

```
#mysql>exit
```

再次登录到 mysql 要用

```
#/usr/local/mysql/bin/mysql -u root -p
```

123456

⑧关闭 mysql 服务

```
#/usr/local/mysql/bin/mysqladmin -u root -p shutdown
```

123456

⑨添加开机自启动

```
#echo "/usr/local/mysql/bin/mysqld_safe --user=mysql &" >>/etc/rc.local
```

```
#cat /etc/rc.local //查看命令是否追加到 rc.local 中
```

## 第三章 Apache 安装

### 3.1 Apache 安装

①解压并进入

```
#cd /usr/local/src
```

```
#tar zxvf httpd-2.2.9.tar.gz
```

```
#cd http-2.2.9
```

①#./configure \

```
--prefix=/usr/local/apache \      //创建 apache 软件安装的目录
```

```
--enable-so \                      //必选项 以动态共享对象（DSO）编译  
                                //Php 作为 apache 动态对象装进去
```

```
--enable-rewrite
```

②#make

③#make install

### 3.2 测试 apache 服务器

①检查安装目录

```
#ls /usr/local/apache
```

②检查配置文件目录

```
#ls /usr/local/apache/conf
```

③启动 apache

```
#!/usr/local/apache/bin/apachectl start
```

如果出现错误 httpd: Could not reliably determine the server's fully qualified domain name, using  
localhost.localdomain for ServerName

解决办法：只要把/conf/httpd.conf 文件的#ServerName 前的#去掉，然后重启下服务

④查看 80 端口是否开启

```
netstat -tnl|grep 80
```

也可以查看进程：

```
#ps -le|grep httpd
```

⑤关闭 apache

```
#!/usr/local/apache/bin/apachectl stop
```

⑥远程访问 apache 服务器

ifconfig 查看 IP 地址，找到 httpd.conf 配置文件修改 ServerName 为本机 IP

```
#cd /usr/local/apache/conf/httpd.conf
```

```
#vi httpd.conf
```

修改 ServerName 192.168.1.109:80

修改配置文件后要将服务重启后才生效。

远程登录输入网址格式 http://IP/



这时候如果远程访问不了，将服务器防火墙关掉

```
# /etc/init.d/iptables status      //查看防火墙状态
```

```
# /etc/init.d/iptables stop        //关闭防火墙
```

如果网页显示 It Works! 说明 apache 安装成功

⑦添加自启动

```
echo "/usr/local/apache/bin/apachectl start" >>/etc/rc.local
```

## 第四章 PHP 安装

### 4.1 Php 库文件安装

在安装 php 之前，有时候我们需要安装 php 所需要的库文件。4.1php 库文件的安装主要是为了以后 php 的应用，我们可以根据项目需要有选择的安装。

#### 1. 安装 libxml2 最新库文件

```
#cd /usr/local/src/libxml2-2.6.30
```

```
①./configure --prefix=/usr/local/libxml2
```

```
②make
```

```
③make install
```

其中①prefix 参数指定安装目录为 libxml2，如果该目录不存在，则在配置阶段会自动创建，安装过程中注意每个步骤完成后有无 error 错误

库文件的安装成功检测：

进入安装后的目录：①cd /usr/local/libxml2/

②ls 后看有没有一些子目录

```
sudo apt-get install libxml2  
sudo apt-get install libxml2-dev
```

#### 2. 安装 libmcrypt 最新库文件

```
#cd /usr/local/src/libmcrypt-2.5.8
```

```
①./configure --prefix=/usr/local/libmcrypt
```

```
②make
```

```
③make install
```

```
#cd /usr/local/src/libmcr-2.5.8/libltdl
```

```
②#./configure --enable-ltdl-install
```

```
③#make
```

```
④#make install
```

#### 3. 安装 zlib 最新库文件

直接安装在系统库里，不用指定安装目录

```
#cd /usr/local/src/zlib-1.2.3
```

```
①./configure
```

```
②make
```

```
③make install
```

#### 4. 安装 libpng 最新库文件（zlib 要在之前安装）

```
#cd /usr/local/src/libpng-1.2.31
```

```
①./configure --prefix=/usr/local/libpng
```

②make

③make install

## 5.安装 jpeg6 最新库文件

安装目录不会自动创建，需要自己创建

```
mkdir /usr/local/jpeg6 //建立 jpeg6 软件安装目录
```

```
mkdir /usr/local/jpeg6/bin //建立存放命令的目录
```

```
mkdir /usr/local/jpeg6/lib //建立 jpeg6 库文件所在目录
```

```
mkdir /usr/local/jpeg6/include //建立存放头文件目录
```

```
mkdir -p /usr/local/jpeg6/man/man1 //建立存放手册的目录
```

```
#cd /usr/local/src/jpeg-6b
```

①./configure --prefix=/usr/local/jpeg6 \

--enable-shared \//建立共享库使用的 GNU 的 libtool

--enable-static //建立静态库使用的 GNU 的 libtool

②make

③make install

有时会编译时会出现“make:libtool:命令未找到”错误

解决办法：首先看有没有安装 libtool

```
#rpm -qa|grep libtool
```

然后将 libtool 里的两个配置文件覆盖掉 jpeg-6b 里的配置文件(cent OS 6.2 系统则较下多了一级 config 文件夹)

```
#cp /usr/share/libtool/config.sub /usr/local/src/jpeg-6b/config.sub 回车 y
```

```
#cp /usr/share/libtool/config.guess /usr/local/src/jpeg-6b/config.guess 回车 y
```

然后重新 configure

## 6.安装 freetype 最新库文件

```
#cd /usr/local/src/freetype-2.3.5
```

① ./configure --prefix=/usr/local/freetype

②make

③make install

## 7.安装 autoconf 最新库文件

直接安装在系统库里，不用指定安装目录

```
#cd /usr/local/src/autoconf-2.61
```

①./configure

②make

③make install

## 8.安装最新的 GD 库文件

```
#cd /usr/local/src/gd-2.0.35
```

```
①./configure --prefix=/usr/local/gd2 \ //创建软件安装的位置
--with-zlib=/usr/local/zlib \ //指定到哪里去找 zlib 库文件的位置，如果安装
zlib 时没有指定安装目录，则此配置选项不用写，会默认安装
-- with-jpeg=/usr/local/jpeg6 \
-- with-png=/usr/local/libpng \
-- with-freetype=/usr/local/freetype
```

```
②make
```

```
② make install
```

## 9.安装 curl

```
#cd /usr/local/src/curl-7.17.1
```

```
① ./configure --prefix=/usr/local/curl
```

```
② make
```

```
③ make install
```

## 4.2 PHP 安装

因为 php 是作为 apache 的配置模块安装的，所以启动 apache 即启动了 php

```
①#./configure \
```

```
--prefix=/usr/local/php \
```

```
--with-apxs2=/usr/local/apache/bin/apxs \ //必选项 告诉 php 查找 Apache 的地方
```

```
--with-mysql=/usr/local/mysql \ //必选项
```

```
--with-curl=/usr/local/curl
```

```
--with-libxml-dir=/usr/local/libxml2 \
```

```
--with-png-dir=/usr/local/libpng \
```

```
--with-jpeg-dir=/usr/local/jpeg6 \
```

```
--with-freetype-dir=/usr/local/freetype \
```

```
--with-gd=/usr/local/gd2 \
```

```
--with-mcrypt=/usr/local/libmcrypt \
```

```
--with-mysqli=/usr/local/mysql/bin/mysql_config \
```

```
--enable-soap \
```

```
--enable-mbstring=all \
```

```
--enable-sockets
```

```
②make virtual memory exhausted: Cannot allocate memory
```

```
③make test ---- --disable-fileinfo
```

```
④make install
```

make 的时候出现以下错误/usr/bin/ld: cannot find -lldl

collect2: ld returned 1 exit status

make: \*\*\* [libphp5.la] Error 1

解决办法：这是因为找不到 libltdl 库文件

```
#ldconfig -p|grep ltdl
```

安装 libltdl 库文件

```
#cd /usr/local/libmccry-2.5.8/libltdl
```

```
#./configure --enable-ltdl-install
```

```
#make
```

```
#make install
```

## 4.3 Lamp 环境整合

①查看 php 模块是否已经在 apache 下

```
#cd /usr/local/apache/modules
```

```
#ls
```

是否有 libphp5.so 模块

③ 创建 php 配置文件

```
#cp /usr/local/src/php-5.2.6/php.ini-dist /usr/local/php/etc/php.ini
```

③用 vi 编辑 apache 配置文件

```
#vi /usr/local/apache/conf/httpd.conf
```

添加：AddType application/x-httpd-php .php .phtml 指定后缀名让服务器将解析，可任意添加

修改：DirectoryIndex index.php index.html

对配置文件所做的修改，需要重启服务，又因为 php 作为 apache 的加载模块，所以在修改 php 配置文件后也要重启 apache 服务

```
#/usr/local/apache/bin/apachectl stop
```

```
#/usr/local/apache/bin/apachectl start
```

④

```
#cd /usr/local/apache/htdocs
```

编辑#vi test.php

```
<?php
```

```
    Phpinfo();
```

```
?>
```

在网页上输入 http://192.168.1.109/test.php

⑤再次修改 test.php 添加数据库相关代码测试是否连接 mysql

```
<?php
```

```
    If(mysql_connect('localhost','root','123456'))
```

```
{
```

```
    Echo'OK';
```

```
}
```

```
Else
```

```
{
```

```
Echo'Failed';})?>
```

⑥以后做好的 php 网页放到/usr/local/apache/htdocs 下，如果是文件夹，要将文件夹的用户改为运行 apache 用户（有些需要写文件）

```
#ps -aux|grep httpd    //查看运行 apache 用户
```

```
#chown -R daemon.daemon  上传的文件夹名
```

## 第五章 总结

### 5.1 apache、mysql、php 配置文件

①apache 安装时如果没指定配置文件的目录，安装后默认在

```
#vi /usr/local/apache (安装路径) /conf/httpd.conf
```

②mysql 和 php 安装后未生成配置文件，需要我们自己从源码包创建到指定目录

```
#cp /usr/local/php-5.2.6/ php.ini-dist /usr/local/php/etc/php.ini
```

```
#cp /usr/local/mysql-5.0.41/support-files/my-medium.cnf /etc/my.cnf
```

③修改配置文件后，要重启服务才会生效。其中 php 配置文件修改后，重启的是 apache 服务

```
Apache: /usr/local/apache/bin/apachectl start
```

```
        /usr/local/apache/bin/apachectl stop
```

```
MySQL: /usr/local/mysql/bin/mysqld_safe &
```

```
        /usr/local/mysql/bin/mysqladmin -u root -p shutdown
```

④查看服务是否开启

```
#Netstat -tnl //查看 80 端口和 3306 端口是否打开
```

⑤为 apache 和 mysql 添加开机自启动

```
echo"/usr/local/mysql/bin/mysqld_safe --user=mysql &">>/etc/rc.local
```

```
echo"/usr/local/apache/bin/apachectl start">>/etc/rc.local
```

### 5.2 配置文件常见参数修改

①Apache:/usr/local/apache/conf/httpd.conf

```
ServerName: 192.168.1.109 //改为本机 IP
```

```
DirectoryIndex index.php index.html //增加 php 默认网页
```

```
Addtype application/x-httpd-php .php //添加 php 后缀名让服务器能够解析 php 文件
```

### 5.3 安装常见问题解决

①在安装过程中出现错误想重新安装时要注意哪些问题？

答：对于 php 库文件来说，直接执行

```
#rm -rf 安装文件夹名 //直接删除安装目录
```

```
#rm -rf 解压后的源代码包名
```

这样，再重新解压编译安装

对于 apache、mysql 和 php 来说，要先把进程停掉后再删除所有的安装文件夹和解压后的源代码包，然后重新解压编译安装

②如果在安装过程中没有指定安装目录，以后想卸载的时候到哪里去找呢？

答：对于没有指定安装目录的软件安装，我们可以在 make install 的时候做一个安装日志

```
#make install >/backup/zlib/20120519.install.log
```

这样以后在卸载软件时，就能找到相应的安装目录了

③为什么安装完 mysql 后使用 service mysqld start 命令显示未被识别的服务？

答：这是因为 mysql 的服务还没有被添加到 Linux 服务上，添加方法

```
#cp /usr/local/src/mysql-5.0.41/support-files/mysql.server /etc/rc.d/init.d/mysqld
```

```
#chown root.root /etc/rc.d/init.d/mysqld
```

```
#chmod 755 /etc/rc.d/init.d/mysqld
```

```
#chkconfig --add mysqld
```

```
#chkconfig --list mysqld
```

```
#chkconfig --levels 245 mysqld off
```

④为什么安装完 apache 后使用 service httpd start 命令显示未被识别的服务？

答：这是因为 httpd 还没有添加到 Linux 上，添加方法：

```
#cp /usr/local/apache/bin/apachectl /etc/rc.d/init.d/httpd
```

```
#chown root.root /etc/rc.d/init.d/httpd
```

```
#chmod 755 /etc/rc.d/init.d/httpd
```

## 5.4 补充安装

①加速器 ZendOptimizer 的安装

作用：加快 php 访问速度

```
#cd /usr/local/src/
```

```
#tar zxvf ZendOptimizer-3.2.6-linux-glibc21-i386.tar.gz
```

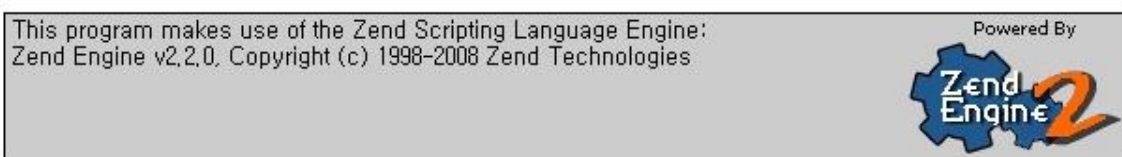
```
#cd ZendOptimizer-3.2.6-linux-glibc21-i386
```

#./install.sh 后按照提示步骤来安装，会提示 Zend 的安装位置，php 配置文件的位置以及 apache 启动命令的位置等。

安装完后，php 的配置文件的位置改到了 Zend 的安装位置，即/usr/local/Zend/etc/php.ini 同时原始配置文件在原来的位置做了备份：/usr/local/php/etc/php.ini-zend\_optimizer.bak

测试是否安装成功：

查看 phpinfo()生成的网页，有 zend 选项则代表安装成功



④ phpMyAdmin

作用：管理数据库的 web 页面，可以远程管理服务器上的数据库

```
#cd /usr/local/src
```

```
#tar zxvf phpMyAdmin-3.0.0-rc1-all-languages.tar.gz
```

//拷贝目录到指定位置并改名为 phpmyadmin

```
#cp -a phpMyAdmin-3.0.0-rc1-all-languages /usr/local/apache/htdocs/phpmyadmin
```

```
#cp /usr/local/apache/htdocs/phpMyAdmin/config.sample.inc.php config.inc.php//生成配置文件
```



phpMyAdmin 的配置: #vi config.inc.php 修改 auth\_type 选择就可以改变身份验证模式, 默认情况为 cookie 身份验证模式。

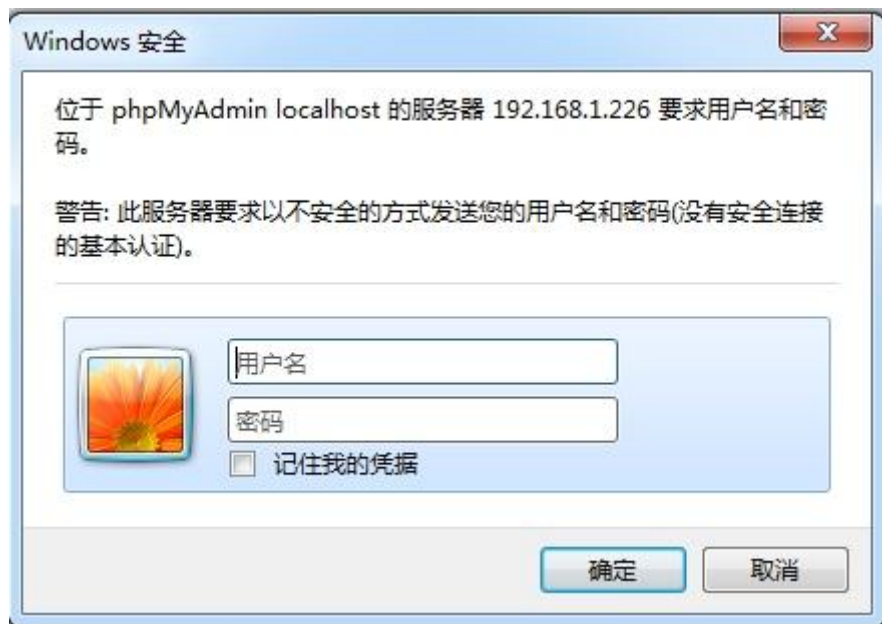
http 和 cookie 身份验证模式要求用户在 phpMyAdmin 登录窗口里输入 MySQL 数据库有效用户名和密码, 才能使用 phpMyAdmin 程序。这两种身份模式尤其适合数据库中多个用户账号的情况。

config 身份验证模式, 这种模式与前两种不同的是, 要将 MySQL 用户名和密码写 config.inc.php 文件中, 这样在登录 phpMyAdmin 时就不会提示输入用户名和密码了, 如果只是在本地系统上使用 phpMyAdmin, 就可以使用这种模式。

修改方式:

② http 身份验证模式: \$cfg['Servers'][\$i]['auth\_type'] = 'http'; // 直接改为 http 即可。

② cookie 身份验证模式: \$cfg['blowfish\_secret'] = 'xxxxxx'; // 这里需要一个任意字符串  
\$cfg['Servers'][\$i]['auth\_type'] = 'cookie'



③ config 身份验证模式: \$cfg['Servers'][\$i]['auth\_type'] = 'config';

// 添加以下两个选项

\$cfg['Servers'][\$i]['user'] = 'root'; // MySQL 数据库用户名

\$cfg['Servers'][\$i]['password'] = '123456'; // 数据库密码

## 5.5 ./configure --help

在配置阶段如果不确定配置选项参数怎么写, 我们在进入解压后的包后执行 ./configure --help 命令

#!/configure --help|grep apxs2 (关键词)