

## 1. 先安装 MySQL-4.1.20

用的是 MySQL4.1.tar.gz 这个 Binary 包，解包后直接就可以运行 MySQL

```
[root@localhost root]# cp MySQL4.1.tar.gz /usr/local/
[root@localhost root]# groupadd mysql
[root@localhost root]# useradd -g mysql mysql
[root@localhost root]# cd /usr/local
[root@localhost local]# gunzip < MySQL4.1.tar.gz | tar xvf -
[root@localhost local]# ln -s mysql-standard-4.1.20-pc-linux-gnu-i686 mysql
[root@localhost local]# cd mysql
[root@localhost mysql]# scripts/mysql_install_db --user=mysql
[root@localhost mysql]# chown -R root .
[root@localhost mysql]# chown -R mysql data
[root@localhost mysql]# chgrp -R mysql .
[root@localhost mysql]# bin/mysqld_safe --user=mysql &
[root@localhost mysql]# cd bin
[root@localhost mysql]# ./mysql
Welcome to the MySQL monitor. Commands end with ; or g.
Your MySQL connection id is 1 to server version: 4.1.20-standard
Type 'help;' or 'h' for help. Type 'c' to clear the buffer.
mysql>
```

如果能看到上面的信息，那么 mysql 的安装就 OK 了。

q

离开

```
# ./mysqladmin -u root password '*****' (默认密码为空自己添加一个吧)
# cp /usr/local/mysql/support-files/mysql.server /etc/rc.d/init.d/mysqld
# chown -R mysql:mysql /etc/rc.d/init.d/mysqld
# chkconfig --add mysqld (将 mysql 添加进自启动服务列表里)
# ntsysv (选中 mysqld 则每次系统启动都能启动 mysql)
```

```
<<<<<<<<<//////////////////////////////////// MySQL 注 释 开 始
////////////////////////////////////
```

为了让 mysql 随开机一起启动请修改 /etc/rc.d/rc.local 文件

在 touch /var/lock/subsys/local 下面一行增加

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

卸掉此种安装模式下的 MySQL 数据库的方法

手动删除 /usr/local/mysql 文件夹

MySQL 的 root 密码修改方法

(1). 在 mysql 目录下 killall mysqld 这个命令用来停止 mysql 服务

```
[root@localhost mysql]# killall mysqld
```

(2). 绕过密码检测以安全模式启动 MySQL

```
[root@localhost mysql]# ./bin/mysqld_safe --skip-grant-tables &
```

(3). 完成以后就可以不用密码进入 MySQL 了

```
[root@localhost root]# ./bin/mysql -u root -p
Enter password: 输入新密码
```

(4). 更改密码

```
>use mysql
>update user set password=password("新密码") where user="root";
>flush privileges;
```

补充说明:

给 mysql 设置密码

```
[root@localhost root]# ./bin/mysqladmin -u root password 'new-password'
[root@localhost root]# ./bin/mysqladmin -u root -h sa password 'new-password'
```

改 mysql 密码的另一个方法

```
./bin/mysqladmin -u root -p 旧密码 新密码
```

再输入一次 旧密码

```
//////////////////////////////////// MySQL 注 释 结
束 //////////////////////////////////>>>>>>>>
```

2. 安装 APACHE-2.2.2

```
[root@localhost root]# tar zvxf httpd-2.2.2.tar.gz
```

```
[root@localhost root]# cd httpd-2.2.2
[root@localhost httpd-2.2.2]# ./configure --prefix=/usr/local/apache2 --enable-so
[root@localhost httpd-2.2.2]# --enable-mods-shared=most --disable-charset-lite
[root@localhost httpd-2.2.2]# make
[root@localhost httpd-2.2.2]# make install
[root@localhost httpd-2.2.2]# cp /usr/local/apache2/bin/apachectl /etc/rc.d/init.d/httpd
[root@localhost httpd-2.2.2]# ln -s /etc/rc.d/init.d/httpd /etc/rc3.d/S97httpd
```

<<<<<<<<<//////////////////////////////////// Apache 注 释 开  
始 //////////////////////////////////////

--enable-so 开启 DSO 支持， --enable-mods-shared=most 是指除了 so 以外的所有标准模块都编译成 DSO 模块， --disable-charset-lite 多国语言支持

启动、停止与重启 Apache 服务

```
[root@localhost root]# cd /etc/rc.d/init.d
[root@localhost init.d]# ./httpd stop
[root@localhost init.d]# ./httpd start
[root@localhost init.d]# ./httpd restart
```

或者

```
[root@localhost root]# cd /usr/local/apache2/bin
[root@localhost bin]# ./apachectl stop
[root@localhost bin]# ./apachectl start
[root@localhost bin]# ./apachectl restart
```

Apache 服务器的自启动:

修改 /etc/rc.d/rc.local 文件

在末尾加上

/usr/local/apache2/bin/apachectl start 这一行。

//////////////////////////////////// Apache 注 释 结  
束 //////////////////////////////////////>>>>>>>>

### 3. GD 库的安装 (Discuz 版 仅支持 PHP-5.1.2)

```
shell> cd /usr/local/src
shell> wget -c http://download.discuz.net/gdinstaller.tar.gz
shell> tar xzvf gdinstaller.tar.gz
```

```
shell> cd GDInstaller
shell> ./install.sh
```

#### 4. 安装 PHP-5.1.2 (DSO)

```
shell> wget -c http://download.discuz.net/php-5.1.2.tar.bz2
shell> tar xjvf php-5.1.2.tar.bz2
shell> cd php-5.1.2
```

```
shell> ./configure --prefix=/usr/local/php --with-mysql=/usr/local/mysql
--with-apxs2=/usr/local/apache2/bin/apxs
--with-zlib-dir --with-bz2 --with-tiff-dir --with-libxml-dir --with-gd --with-freetype-dir
--with-jpeg-dir --with-png-dir --with-ttf --enable-mbstring=all
--with-config-file-path=/usr/local/php/etc --disable-ipv6 --enable-gd-native-ttf
```

( 如果 configure 最后出现以下错误信息: configure: error: libxml2 version 2.6.11 or greater required. 则补充操作如下:

```
shell> wget ftp://ftp.gnome.org/mirror/gnome.org/sources/libxml2/2.6/libxml2-2.6.11.tar.gz
shell> tar -zxvf libxml2-2.6.11.tar.gz
shell> cd libxml2-2.6.11
shell> ./configure --enable-ipv6=no
shell> make
shell> make install
```

再重新退回上级目录并 configure )

```
shell> make
```

```
shell> make install
```

```
shell> cp php.ini-recommended /usr/local/php/etc/php.ini
```

编辑 php.ini

修改如下参数变量

register\_globals = On # 打开自动全局变量, 允许网页之间传递变量

expose\_php = Off # 不显示 PHP 版本号 (可以不设置, 设置了安全点儿)

```
<<<<<<<<<//////////////////////////////////////          php          注          释          开
始 //////////////////////////////////////////////////
```

判断 php 版本的方法:

执行

```
shell> /usr/local/php/bin/php -v
```

来查看版本, 第一行就会说 php 的版本

卸载当前 PHP

```
shell> rm -rf /usr/local/php
```

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

找到 LoadModule php5\_module modules/libphp5.so 这一行, 并将此行删除, 保存退出。

(注意卸载的先后顺序, 如果是已经安装过 Zend Optimizer 的, 那就必须先卸载 Zend 再卸载 PHP)

```
//////////////////////////////////////          php          注          释          结  
束 ////////////////////////////////////////>>>>>>>>
```

## 5. 安装 ZendOptimizer-2.6.2

```
shell> cd /usr/local/src
```

```
shell> wget -c http://download.discuz.net/ZendOptimizer-2.6.2-linux-glibc21-i386.tar.gz
```

```
shell> tar xzvf ZendOptimizer-2.6.2-linux-glibc21-i386.tar.gz
```

```
shell> cd ZendOptimizer-2.6.2-linux-glibc21-i386
```

```
shell> ./install.sh
```

\*注意: 安装 Zend 过程中提示输入 php.ini 位置时, 请填写/etc (如不成功 则填写 /usr/local/php/etc 肯定有一个对)

安装好后, 会提示 The php.ini file has been relocated from /usr/local/php/etc to /usr/local/Zend/etc, 并且原 php.ini 建立连接到新的 php.ini 上。最后要求重启 Apache, 照做就是。

```
<<<<<<<<<//////////////////////////////////////          Zend          注          释          开  
始 ////////////////////////////////////////>>>>>>>>
```

判断 php 有没有 zend 支持的方法:

```
shell> /usr/local/php/bin/php -m
```

如果在[Zend Modules]栏目下有 Zend Optimizer 的字样, 说明已经开了 zend 加速支持

卸载当前 Zend

```
shell> rm -rf /usr/local/Zend
```

```
//////////////////////////////////////          Zend          注          释          结  
束 ///////////////////////////////////>>>>>>>>>
```

## 6. 编辑 httpd.conf

添加如下语句:

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

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

```
AddDefaultCharset gb2312 # 默认中文简体
```

修改如下参数变量:

在 DirectoryIndex index.html 后增加 index.php

```
ServerTokens Prod # 不显示 OS 和版本号
```

重新启动一下 apache

```
shell> /usr/local/apache2/bin/apachectl restart
```

## 7. 测试

测试 PHP, 编辑 php.php, 内容如下

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

有的机器上, 测试 phpinfo.php 输出空白页, 右键选择编码-->简体中文即可; 测试 MYSQL 数据库连接, 编辑 mysql.php, 内容如下

```
<?  
$conn=mysql_pconnect("localhost","","");  
if ($conn) echo "ok";  
?>
```

如果网页可以正常显示, 说明 Mysql4.1.20+Apache2.2.2+GD+PHP5.1.2+ZendOptimizer2.6.2 安装配置成功。

环境配置好之后, 开始控制软件的安装与设置:

## 8. 安装 Webmin-1.270 控制服务器

```
shell> wget -c
http://switch.dl.sourceforge.net/sourceforge/webadmin/webmin-1.270-1.noarch.rpm
shell> rpm -Uvh webmin-1.270-1.noarch.rpm
```

安装完毕后用 `http://yourIPAddress:10000` 登陆，登陆帐号密码同服务器

## 9. 设置 FTP 服务器

由于在安装 Linux 时已选择了 RH Linux 9 自带的 VSFTP，默认状态下，该 FTP 是可以匿名登陆的，但是只能下载不能上传文件，

我们需要对这个 FTP 服务做进一步的配置。

下面将创建一个能够满足常用需求的 FTP 服务器。实际应用中，FTP 服务器一般要同时提供上传和下载功能。此外，出于安全考虑，还需要有用户身份验证、用户权限设置及空间管理等。下面就来搭建这样一个 FTP 服务器。

1. 创建欢迎语。如果希望使用者在进入目录时，能够看到欢迎语或对本目录的介绍，可以通过以下方法来实现。确定 `/etc/vsftpd/vsftpd.conf` 文件中 `dirmessage_enable=YES`，默认情况下，Red Hat9.0 有此设置。接着，在目录中新增名为 `.message` 的文件。本例在 `/home/ylg` 目录下创建一个 `.message` 文件，其内容为“欢迎来到我的 FTP 站点”。

2. 更换 FTP 服务器的默认端口。将预设的 21 端口改为 2121，这样做是基于安全的考虑。更改方法为，使用 `vi` 打开 `/etc/vsftpd/vsftpd.conf`：

```
#vi /etc/vsftpd/vsftpd.conf
```

在文件最后增加如下一行内容：

```
listen_port=2121
```

3. 取消 anonymous 登录的功能。在 `vsftpd.conf` 文件中找到如下一行，并将其值改为“NO”：  
`anonymous_enable=YES`

4. 增加一个叫做 star 的 FTP 用户，并将其建立于 `/usr/local/apache2/htdocs/news` 目录下，该用户仅能 FTP 访问服务器，不具有其他权限。

```
shell> adduser -d /usr/local/apache2/htdocs/news/star -g ftp -s /sbin/nologin star
shell> passwd star
Changing password for user star.
New password: ***** (不可见)
Retype new password: ***** (不可见)
```

```
passwd: all authentication tokens updated successfully.  
shell>
```

这个方式建立的用户仍能够看到上级目录的文件，虽然不可以上传，但仍具有下载权限，那么，为了将其限定在初始目录里，不使其浏览上一级目录或其它目录，我们需要进行如下操作：我们可以通过更改 `vsftpd.conf` 文件，加入如下的一行

```
chroot_local_user=YES
```

改完配制文件，不要忘记重启 vsFTPd 服务器

```
shell> /etc/init.d/vsftpd restart  
关闭 vsftpd: [ 确定 ]  
为 vsftpd 启动 vsftpd: [ 确定 ]  
shell>
```

附：增加和删除一个系统用户的方法

增加一个系统用户

```
shell> useradd eurostar
```

删除之

```
shell> userdel -r eurostar
```

增减的用户在 `/home` 目录下会建立或删除相应的文件夹

4. 设定使用者不得更改目录。这样做的目的也是基于安全性的考虑。一般情况下，使用者的预设目录为 `/home/username`。若是不希望使用者在登录后能够切换至上一层目录 `/home`，则可通过以下设置来实现。在 `/etc/vsftpd/vsftpd.conf` 文件中找到以下三行内容：

```
#chroot_list_enable=YES  
# (default follows)  
#chroot_list_file=/etc/vsftpd.chroot_list
```

将其改为：

```
chroot_list_enable=YES  
# (default follows)  
chroot_list_file=/etc/vsftpd/chroot_list
```

（就是去掉前面的#号嘛，不要看花了眼 :)）



新增一个文件/etc/vsftpd/chroot\_list，文件内容为两个用户名：

```
ylg  
user1
```

5. 针对不同的使用者限制不同的速度。假设用户 ylg 所能使用的最高速度为 500Kb/s，用户 user1 所能使用的最高速度为 250Kb/s，可以通过以下方法设置。在/etc/vsftpd/vsftpd.conf 文件尾部新增以下一行：

```
user_config_dir=/etc/vsftpd/userconf
```

增加一个名为/etc/vsftpd/userconf 的目录：

```
#mkdir /etc/vsftpd/userconf
```

在/etc/vsftpd/userconf 下新增一个名为 ylg 的文件，其内容如下所示：

```
local_max_rate=500000
```

在/etc/vsftpd/userconf 目录下新增一个名为 user1 的文件，其内容如下所示：

```
local_max_rate=250000
```

VSFTP 对于速度的限制范围大概在 80%到 120%之间，也就是限制最高速度为 100Kb/s，但实际的速度可能在 80Kb/s 到 120Kb/s 之间。如果频宽不足，数值会低于此限制。

6. 对于每一个联机用户，都以独立的进程来运行。一般情况下，在启动 VSFTP 时，只会看到一个名为 vsftpd 的进程在运行。但若是读者希望每一个联机用户都能以独立的进程来呈现，则可通过在/etc/vsftpd/vsftpd.conf 文件中增加以下一行来实现：

```
setproctitle_enable=YES
```

7. 保存/etc/vsftpd/vsftpd.conf 文件，然后重新启动 vsftpd:

```
#service vsftpd restart
```

8. 测试刚创建的 FTP 服务器。

以缺省方式登录会被拒绝，因为此时的默认端口号已经更改为 2121，所以登录时需指定端口。

```
# ftp 127.0.0.1  
ftp: connect: Connection refused
```

此时也不能再使用匿名方式登录：

```
# ftp 127.0.0.1 2121
Connected to 127.0.0.1 (127.0.0.1).
220 (vsFTPD 1.1.3)
Name (127.0.0.1:root): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
```

如果以用户 ylg 则可以成功登录（指定端口 2121），并显示欢迎信息：

```
# ftp 127.0.0.1 2121
Connected to 127.0.0.1 (127.0.0.1).
220 (vsFTPD 1.1.3)
Name (127.0.0.1:root): ylg
331 Please specify the password.
Password:
230-欢迎来到我的 FTP 站点
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
```

因为在设置中设定了不能切换目录，所以下列命令无法正确执行：

```
ftp> cd /home
550 Failed to change directory.
```

再来测试一下上传和下载。首先下载服务器目录中的 test.txt 文件：

```
ftp> get test.txt
local: test.txt remote: test.txt
227 Entering Passive Mode (127,0,0,1,243,215)
150 Opening BINARY mode data connection for test.txt (21 bytes).
226 File send OK.
21 bytes received in 0.00308 secs (6.7 Kbytes/sec)
```

可以通过!ls 命令看到本机目录中已成功下载该文件。然后上传本机目录中的 ylg.txt 文件到服务器：

```
ftp> put ylg.txt
local: ylg.txt remote: ylg.txt
227 Entering Passive Mode (127,0,0,1,133,248)
150 Ok to send data.
```

```
226 File receive OK.  
19 bytes sent in 0.0401 secs (0.46 Kbytes/sec)
```

用 ls 命令查看服务器目录，会发现该文件已成功上传。

为了测试不同连机用户使用的是不同进程，可以使用 ps -ef 指令，显示如下所示：

```
# ps -ef|grep ftp  
root 12972 1356 0 13:44 pts/1 00:00:00 ftp 127.0.0.1 2121  
nobody 12973 12908 0 13:44 ? 00:00:00 [vsftpd]  
yhg 12975 12973 0 13:44 ? 00:00:00 [vsftpd]  
user1 13013 13011 0 13:46 ? 00:00:00 [vsftpd]  
root 13041 13015 0 13:47 pts/4 00:00:00 grep ftp
```

到现在为止，一个基本可以满足普通使用需求的 FTP 服务器就已经架设完成。

在实际应用中，有时为了增加安全性，会将 FTP 服务器置于防火墙之后。如本文开头所述，被动传输模式适合于带有防火墙的情况。下面就来创建一个防火墙后的 FTP 服务器，该服务器 FTP 端口为 2121，数据传输端口为 2020。

执行以下两行指令，只允许 2121 和 2020 端口打开，其余端口关闭：

```
#iptables -A INPUT -p tcp -m multiport --dport 2121,2020 -j ACCEPT  
#iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```

修改/etc/vsftpd/vsftpd.conf 文件，在文本最后添加以下两行：

```
listen_port=2121  
ftp_data_port=2020
```

重新启动 vsftpd:

```
#service vsftpd restart
```

有时希望直接在/etc/hosts.allow 中定义允许或拒绝某一源地址，可以通过以下配置来实现。先确保/etc/vsftpd/vsftpd.conf 中 tcp\_wrappers=YES，Red Hat 9.0 中，这是默认值。重新启动 vsftpd:

```
#service vsftpd restart
```

假设提供 168.192.2.1 和 210.31.8.1 到 210.31.8.254 的连接，则可对/etc/hosts.allow 进行如下设定：

```
vsftpd : 168.192.2.1 210.31.8. : allow
```

ALL : ALL : DENY

### 配置虚拟用户 FTP

上面配置的 FTP 服务器有一个特点，就是 FTP 服务器的用户本身也是系统用户。这显然是一个安全隐患，因为这些用户不仅能够访问 FTP，也能够访问其它的系统资源。如何解决这个问题呢？答案就是创建一个虚拟用户的 FTP 服务器。虚拟用户的特点是只能访问服务器为其提供的 FTP 服务，而不能访问系统的其它资源。所以，如果想让用户对 FTP 服务器站内具有写权限，但又不允许访问系统其它资源，可以使用虚拟用户来提高系统的安全性。在 VSFTP 中，认证这些虚拟用户使用的是单独的口令库文件（pam\_userdb），由可插入认证模块（PAM）认证。使用这种方式更加安全，并且配置更加灵活。下面介绍配置过程。

1. 生成虚拟用户口令库文件。为了建立此口令库文件，先要生成一个文本文件。该文件的格式如下，单数行为用户名，偶数行为口令：

```
#vi account.txt
yhg
1234
zhanghong
4321
gou
5678
```

2. 生成口令库文件，并修改其权限：

```
#db_load -T -t hash -f ./account.txt /etc/vsftpd/account.db
#chmod 600 /etc/vsftpd/account.db
```

3. 新建一个虚拟用户的 PAM 文件。加上如下两行内容：

```
#vi /etc/pam.d/vsftp.vu
auth required /lib/security/pam_userdb.so db=/etc/vsftpd/account
account required /lib/security/pam_userdb.so db=/etc/vsftpd/account
```

4. 建立虚拟用户，设置该用户所要访问的目录，并设置虚拟用户访问的权限：

```
#useradd -d /ftpsite virtual_user
#chmod 700 /ftpsite
```

经过该步骤的设置，/ftpsite 就是 virtual\_user 用户的主目录，该用户也是/ftpsite 目录的拥有者。除 root 用户之外，只有该用户具有对该目录的读、写和执行的权限。

5. 生成一个测试文件。先切换至 virtual\_user 用户身份，然后在/ftpsite 目录下创建一个文件：

```
#su -virtual_user
```

```
$vi /ftpsite/mytest
This is a test file.
$su - root
```

6. 编辑/etc/vsftpd/vsftpd.conf 文件，使其整个文件内容如下所示（去掉了注释内容）：

```
anonymous_enable=NO
local_enable=YES
local_umask=022
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=YES
one_process_model=NO
chroot_local_user=YES
ftpd_banner=Welcom to my FTP server.
anon_world_readable_only=NO
guest_enable=YES
guest_username=virtual_user
pam_service_name=vsftp.vu
```

上面代码中，`guest_enable=YES` 表示启用虚拟用户；`guest_username=virtual` 则是将虚拟用户映射为本地用户，这样虚拟用户登录后才能进入本地用户 `virtual` 的目录 `/ftpsite`；`pam_service_name=vsftp.vu` 指定 PAM 的配置文件为 `vsftp.vu`。

7. 重新启动 VSFTP：

```
#service vsftpd restart
```

8. 以虚拟用户 `gou`（Linux 中并无该账号）进行测试：

```
# ftp 127.0.0.1
Connected to 127.0.0.1 (127.0.0.1).
220 Welcom to my FTP server.
Name (127.0.0.1:root): gou
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
```

测试下载服务器目录中的一个文件 mytest:

```
ftp> get mytest
local: mytest remote: mytest
227 Entering Passive Mode (127,0,0,1,159,19)
150 Opening BINARY mode data connection for mytest (21 bytes).
226 File send OK.
21 bytes received in 0.00038 secs (54 Kbytes/sec)
```

测试上传本机目录中的文件 vsftpd.conf:

```
ftp> !ls
account.db chroot_list k mytest userconf vsftpd.conf
ftp> put vsftpd.conf
local: vsftpd.conf remote: vsftpd.conf
227 Entering Passive Mode (127,0,0,1,117,203)
150 Ok to send data.
226 File receive OK.
4229 bytes sent in 0.00195 secs (2.1e+03 Kbytes/sec)
```

可以看到，使用没有系统账号的虚拟用户可以成功完成上传、下载的工作。但该 FTP 虚拟服务器只允许虚拟用户登录，其它系统用户无法登录，如系统用户 user1 不是虚拟用户，则不能登录该虚拟服务器。

```
# ftp 127.0.0.1
Connected to 127.0.0.1 (127.0.0.1).
220 Welcom to my FTP server.
Name (127.0.0.1:root): user1
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
```

在虚拟 FTP 服务器中，也可以对各个用户的权限进行设置。方法是在/etc/vsftpd.conf 文件中添加如下一行：

user\_config\_dir=用户配置文件目录

然后在用户配置文件目录下创建相应的用户配置文件，比如为上述名为 gou 的用户创建一个配置文件（假设置配置文件目录为/etc/user\_config\_dir）：

```
#vi /etc/user_config_dir/gou
write_enable=NO
```

anono\_upload\_enable=NO

重启 FTP 服务器，这时再使用账号 gou 来登录，就已经没有上传的权限了。