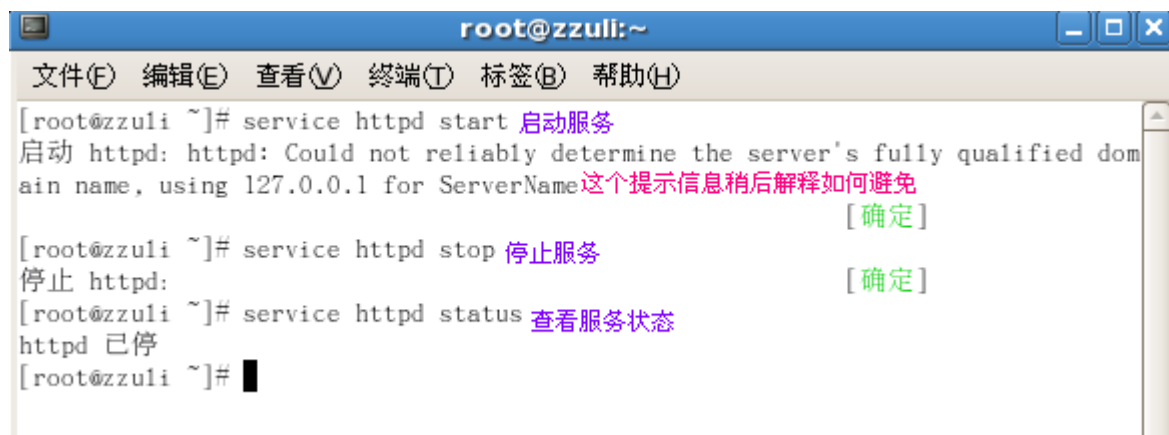


Apache

Apache: 在 1995 年之前就有很多的 web 架设服务器软件的出现, 不过, 真正到了 1995 年之后, 由国际超级计算机应用中心 (NCSA) 主导并克服了一些 Web 主机的 bug 之后, 才让这个 http 协议的 www 套件得到了更广泛的应用! 而因为这个版本是来自于一些 bug 的克服, 因此, 这个 www 套件被称为【A patchy server】, 意思是一个经过更新后的 Server 的意思! 后来, 因为要将名字确定下来, 干脆就直接取其谐音, 用【Apache】。

如果你的 Apache 服务器是默认安装的话, 那么就可以直接用 service 命令来管理 Apache 的启动, 停止, 和参看状态, 如下图所示:



```
root@zzuli:~  
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)  
[root@zzuli ~]# service httpd start 启动服务  
启动 httpd: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName 这个提示信息稍后解释如何避免 [确定]  
[root@zzuli ~]# service httpd stop 停止服务  
停止 httpd: [确定]  
[root@zzuli ~]# service httpd status 查看服务状态  
httpd 已停  
[root@zzuli ~]#
```

注意: 这里要解释一下, 如果避免上面提到的那个信息, 大家都知道 apache 的配置文件在 /etc/httpd/conf 目录下, 你需要更改 apache 的配置文件, 你 vi 打开这个配置文件, 如下图:

```
[root@zzuli ~]# vim /etc/httpd/conf/httpd.conf
```

然后找到 ServerName 这一行:

```
# redirections work in a sensible way.  
#  
#ServerName www.example.com:80
```

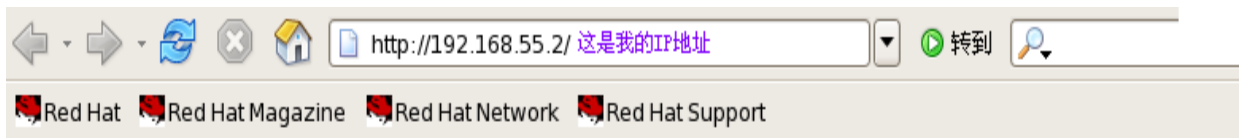
下面你要做的就是把这一行的内容该一下, 就是把 WWW.Example.com 改成你的 IP 地址就可以了, 我的 IP 是 192.168.55.2 所以我的更加就如下图所示:

```
# redirections work in a sensible way.  
#  
#ServerName 192.168.55.2:80
```

别忘了把 ServerName 前面的“#”号去掉。然后你再次启动 Apache 就不会再出现这个信息了:

```
[root@zzuli ~]# service httpd start  
启动 httpd: 已经没有提示信息了 [确定]  
[root@zzuli ~]#
```

当你已经启动了 Apache 的服务之后, 你就可以检测一下 Apache 服务是否正常。这时你启动浏览器, 在浏览器中键入你的 IP 地址, 就会出现 Apache 的测试页面, 如下图所示:



installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

这个页面就是Apache的测试页面

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



其实到这里你就已经可以发布一个简单的网页了，现面我来说明一下怎样发布一个简单的网页：

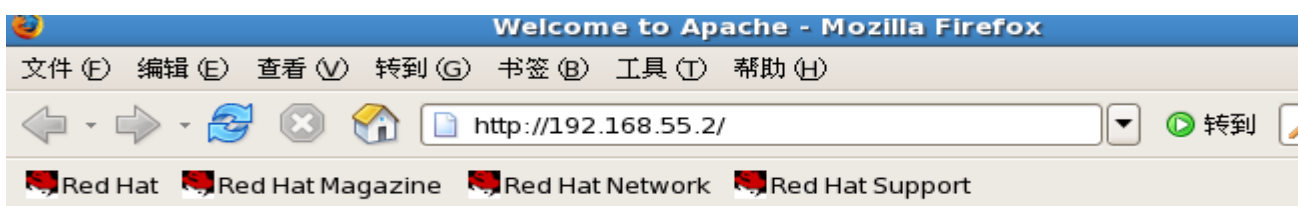
首先：你需要写一个基本的网页，写个很简单的就可以了。

先后：你把这个网页放到 Apache 的默认发布路径上：`/var/www/html` 文件夹

最后：完成上面的工作后你就可以测试了。

```
[root@zzuli ~]# cd /var/www/html/ 默认路径
[root@zzuli html]# vim index.html
[root@zzuli html]# ls
index.html 这个就是要发布的网页
[root@zzuli html]# pwd
/var/www/html 然后就可以测试了
[root@zzuli html]#
```

打开浏览器进行测试，测试结果如下图：



欢迎学习 Apache 服务器

这就是最基本的 Apache 的配置了。

现面我们来介绍一下 httpd.conf 这个文件，请自行参考底下的设定来解决您的 http.conf

此处，需要注意的是基本的环境设定方法为：

<设定项目>

.....

.....

<设定项目>

例如：

<Directory>

Options Index

</Directory>

几乎都是这样的设定。特别留意的是，如果你由额外的设定时，不能随便在 httpd.conf 里头找地方写入，否则如果刚好写在<Directory>.....</Directory>里头，那么就会出现错误了。

下面列出一些基本的 httpd.conf 中的内容：

```
Timeout 300
# 这是用来设定连接到你这部主机的客户端，当超过 300 秒客户端还没有
# 办法连上你的主机时，就予以断线处理！

KeepAlive On
# 是否允许持续性联机，亦即一个联机有多个要求！这里通常设定为 On 比较好，
# 就鸟哥的经验来看，设定为 Off 似乎会产生很多 Time_Wait 的封包！粉怪！

MaxKeepAliveRequests 100
# 在持续性的联机当中，最多允许的联机数目！如果不要限制，可以设定为 0，
# 当然，官方网站上面说，要有较佳的效能，最好设定大一点，所以我都将他改为 200 以上。

KeepAliveTimeout 15
# 同一个联机的 Client 下次的需求没有在 15 秒内送出，那么该联机会被视同断线喔！
```

```
<IfModule prefork.c>
StartServers      5
MinSpareServers   5
MaxSpareServers   10
MaxClients        150
MaxRequestsPerChild 0
</IfModule>
<IfModule worker.c>
StartServers      2
MaxClients        150
MinSpareThreads   25
MaxSpareThreads   75
ThreadsPerChild   25
MaxRequestsPerChild 0
</IfModule>
```

这两段主要是与系统的效能较有关系！如果不需要效能设定的话，那么使用默认值就足够了！

1. MinSpareServers 与 MaxSpareServers 是开启 httpd 服务数目的地方，当你执行

/usr/local/apache/bin/apachectl start 之后，在 shell 下执行

ps -aux|grep http 就可以看到 http 的数量，通常这与你的 RAM 有关。

如果是小站的话，可以设小一点，例如最小设 3 最大设 5 即可！

2. 而 StartServers 则设与 Min 相同即可！

3. 至于 MaxClients 则可以设小一点，因为设定太大很耗系统资源，

而太小则无法让很多人连上来！所以可以设成例如 100

```
Listen 80
```

```
# 设定监听的 port , 如果你要更改 WWW 的 port number , 可以在这里修改, 例如 8080
```

```
User nobody
```

```
Group nogroup
```

```
# 这个是设定 apache 所产生的, 就是刚刚我们上头所设定的 MinSpareServers , 之后会产生
```

```
# 一些 process , 那么这些 processes 的拥有者与拥有群组 ( owner & group ) 是谁!
```

```
# 这个与未来的「PID 权限及 Linux 权限设定」有关! 通常如果是 RPM 安装的话,
```

```
# 大致上都会是 apache , 而如果是 Tarball 安装, 通常是 nobody 与 nogroup 吧!
```

```
# 是否有该 user 还要查看 /etc/passwd 及 /etc/group 喔! 不能设错! 否则无法启动 apache
```

```
ServerAdmin root@localhost
```

```
# 这个是设定你的机器的 httpd 管理员账号! 设成你的 e-mail 吧! 例如我都设定为:
```

```
# 我的机器上: ServerAdmin test@localhost
```

```
#ServerName new.host.name:80
```

```
# 设定主机名称的地方, 若有需要的话才设定, 否则可以将他 mark 掉也没关系!
```

```
# 但如果你需要让 Apache 自动帮你将其它名称连过来的主机名称修改时(下一个设定),
```

```
# 那么这里就需要填写啰! 此外的此外, 需要了解的是, 有时, 例如 Openlinux server 3.1.1
```

```
# 如果没有设定这个的话, 那么你的 WWW 将无法启动! 另外, 如果设定错误, 同样无法启用!
```

因此，没有特别要求的话，那么这个就暂时不要设定吧！免得自找麻烦～

UseCanonicalName Off

主机的别名啦！例如你的主机有三个名称时，那么这个 Off 的设定，会让 Client 端

可以分别使用三个名称显示在他们的浏览器上面，如果是 On 的话，那么将以上面的

ServerName 内容来显示在他们的浏览器上面，而不是原来他们写的主机名称喔！

说是这样说，但是我试不出来这个功能～～ @@

AddDefaultCharset ISO-8859-1

LanguagePriority en da nl et fr de el it ja ko no pl pt pt-br ltz ca es sv tw

Apache 的预设显示语言编码！请特别注意，因为这里的设定并不适合台湾的繁体中文，

呵呵！所以底下请【务必】修改成这样！否则您的网页总是无法显示中文喔！

这两行大概在 httpd.conf 的 750 ~ 800 行之间！请以搜寻的方式找一下！

AddDefaultCharset Big5

LanguagePriority tw en da nl et fr de el it ja ko no pl pt pt-br ltz ca es sv

HostnameLookups Off

在记录档案的时候，登录档的内容，来提取我们数据的主机是以 IP 还是主机名称来显示？

当然是 IP 来显示比较快喔！所以，这里通常设定为 Off，不需要转译 IP 成为主机名！

```

[root@test root]# cd /usr/local/apache2/conf
[root@test root]# vi httpd.conf

DocumentRoot "/usr/local/apache2/htdocs"
# 将 /usr/local/apache2/htdocs 设定为 Apache 的根目录！
# 这个就是主机的主网页啦！你可以将他移到任何你高兴的地方！
# 不过，比较重要的限制是，最好这个目录底下不要包含重要的信息，例如你不要将根目录 /
# 设定为这个 DocumentRoot 吧！ ^_^ 否则你的主机下的任何数据，不就任何人都可以使用
# 浏览器来查看？岂不是很危险！

<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
# 这个是设定根目录，亦即是 /usr/local/apache2/htdocs 这咚咚的属性啦！
# 使用 Options 来设定相关属性，相关的属性还有底下几个ㄟ！
ExecCGI      ：使该目录底下的 CGI 具有可以执行的能力！重要项目！如果您要您的
-----某个目录可以执行 CGI 的程序时，那么请将该目录多加 ExecCGI 这个属性！-----
FollowSymLinks  ：让您的 link 的目录或档案，虽然在其它的目录下，仍然可以连接出去！
                  举个例子来说，目前我的主页是 /usr/local/apache2/htdocs ，但我想

```

其他的我就不在这里列举了，请大家自己找一些资料看看。

认证和授权

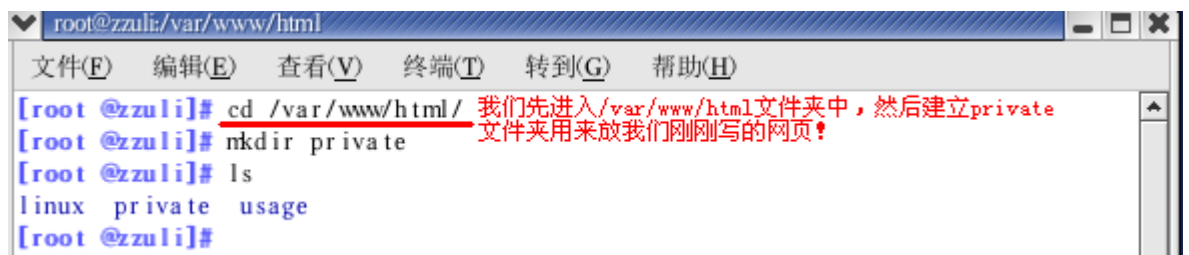
所谓的认证和授权就是，只有拥有密码的人才可以查看该网站的内容。

在这个实验中我们在写一个网站，当然是写一个最简单的，其实就是一个网页，该网页的具体实现，如下图：



```
<html>
  <title>认证和授权</title>
  <body>
    <center>只有拥有用户名和密码的人才能够进入本站点!</center>
  </body>
</html>
```

这是一个非常简单的网页。这就不作太多解释了。
我们要想发布这个网页，就需要把它放在/var/www/html 中。



```
[root @zzuli]# cd /var/www/html/
[root @zzuli]# mkdir private
[root @zzuli]# ls
linux  private  usage
[root @zzuli]#
```

这个网页的名字叫做 index.html。
然后就是重点的内容了，我们需要对这个网页进行认证机制。
当然这种认证机制的实现同样也是需要更改 Apache 的配置文件的。需要在配置文件中添加下图所示的内容：


```
root@zzuli:/var/www/html/private
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)

Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *>
#   ServerAdmin webmaster@dummy-host.example.com
#   DocumentRoot /www/docs/dummy-host.example.com
#   ServerName dummy-host.example.com
#   ErrorLog logs/dummy-host.example.com-error_log
#   CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
<Directory "/var/www/html/private">
#       不使用 .htaccess 文件
#       AllowOverride None
#       指定基本认证方式
#       AuthType Basic
#       指定认证领域名
#       AuthName "zhou"
#       指定认证口令文件的存放位置
#       AuthUserFile /var/www/passwd/zhou
#       授权给认证口令中的所有用户
#       require valid-user
</Directory>

1054,13 底端
```

这一部分就是要加入到配置文件中的内容，在实验的时候自己在配置文件中找个位置添加一下就可以了！
注释比较详细，这里就不多做解释了！

下面我们需要制作口令文件：



```
root@zzuli:/var/www/passwd
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)

[root @zzuli]# mkdir /var/www/passwd 创建passwd目录
[root @zzuli]# cd /var/www/passwd/
[root @zzuli]# htpasswd -c zhou user 用htpasswd命令创建用户user，-c 后面
New password: 的zhou为口令文件名
Re-type new password:
Adding password for user user
[root @zzuli]# chown apache.apache zhou 'zhou' 更改文件的所属权，把
[root @zzuli]# ls 组用户
zhou
[root @zzuli]# ll
总用量 4
-rw----- 1 apache apache 19 6月 1 19:41 zhou 查看文件属性
[root @zzuli]#
```

所有以上工作完成以后，需要重新启动 Apache 服务器。然后我们打开浏览器进行验证：

停止 <http://192.168.55.10/private> 输入网站的地址

Network [Support](#) [Shop](#) [Products](#) [Training](#)



这里就出现了用户登陆的信息，只要你填写正确就可以看到网站的内容了

Red Hat

第一步 激活您的

购买 Red Hat 您能够得到诸
技术支持和对
的使用。激活
获得这些权益

点击这里来开

提示

请为 "zhou"@192.168.55.10输入用户名和密码

用户名:

密码:

☐ 使用密码管理器来记住这些值。

大程度地利用

最新状态。
用户管理

得优选服

便可以订阅 Red Hat 网络。详情请参见 [Red Hat 网络服务](#)。

当输入用户名和密码只有，如果输入正确会出现下面的内容：

[Support](#) [Shop](#) [Products](#) [Training](#)

只有拥有用户名和密码的人才能够进入本站点！

Apache 上的虚拟主机

虚拟主机是指在同一台服务器上实现多个 Web 站点，要想实现虚拟主机功能只能通过如下途径：

- (1) 不同的虚拟主机使用不同的 IP 地址和端口号。即若虚拟主机使用相同的 IP 地址，则必须使用不同的端口号，但这种情况下，客户浏览该服务器时必须指定特定的端口号，否则无法访问。称这种虚拟主机为基于 IP 的虚拟主机。
- (2) 不同的虚拟主机使用不同的主机头。主机头使得在一台计算机上可维护多个域名，实现多个虚拟主机。使用主机头需要配置 DNS 服务器，即设置一台计算机的 IP 地址映射到多个域名。主机头只能在 HTTP/1.1 协议上运行，因此并非所有浏览器都支持主机头。

我们这里主要讲解第一种实现，由于第二种需要配置 DNS 服务器，所以有兴趣的同学可以自己搜一些资料。

下面我们来配置基于 IP 的虚拟主机。

在配置基于 IP 的虚拟主机的时候我们需要为虚拟主机写一个配置文件，然后在 Apache 的配置文件包含虚拟主机的配置文件，这需要在 Apache 的配置文件中假如 Include 语句，如下图所示：

我们在apache的配置文件中加入下面这句话即可，‘vhost-conf.d’为虚拟主机配置文件所在的目录

```
include vhost-conf.d/*.conf
"/etc/httpd/conf/httpd.conf" [已转换] 1052L, 35192C 1052,1 底端
```

然后我就要自己动手写虚拟主机的配置文件了，在写配置文件之前我们需要先建立‘vhost-conf.d’文件夹：

```
[root @zzuli]# cd /etc/httpd/          先进入/etc/httpd目录下，然后建立
[root @zzuli]# mkdir vhost-conf.d      vhost-conf.d文件夹
[root @zzuli]# ls
conf  conf.d  logs  modules  run  vhost-conf.d
```

然后在这个目录下，创建配置文件，配置文件的文件名为‘ip-vhost.conf’，把下图的内容输入即可：

DNS

Dns 的简单架设

这个架设平台是 Redhat 企业版 5，可能在其他的 linux 版本上配置的一些细节不一样。

跟 apache 一样，DNS 的配置文件名为 named.conf，但是在本实验平台上这个文件是不存在的是要自己建立的，它的默认建立路径是/var/named/chroot 目录下（不同版本的 bind 是不一样的）。如下图所示：

```
[root@zzuli etc]# ls
localtime named.conf rndc.key 这个named.conf文件是需要自己建立的
[root@zzuli etc]# pwd
/var/named/chroot/etc 这个就是放置named.conf目录
[root@zzuli etc]#
```

下图是我写的 named.conf 的内容：



```
root@zzuli:/var/named/chroot/etc
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
//设定整体的主机规划！重点在direcotry的意义！
options {
    directory "/var/named";
    //这个是在规定 [我的正反解档案放置的目录]
    allow-query{ any; }; //是否允许他人查询
    allow-transfer{ none; };
};
zone "localhost" { //这个zone代表设定档的预设domain
    type master;
    file "named.localhost";
};
zone "apache.com" { //这是我的zone的域名apache.com
    type master;
    file "named.apache.com";
};
~
~
~
~
~
~
"named.conf" 16L, 453C 3,4-18 全部
```

这里有一点要解释一下，上面我写的 directory 是“/var/named”但是当我把 named.localhost 文件和 named.apache.com 文件放如这个目录下的时候回出现错误（无法启动 DNS 服务）。只有把这两个文件放到下图所示的路径下才可以？（这里我也不知道是怎么回事？）

```
[root@zzuli named]# ls
data named.apache.com named.localhost slaves 两个域名设置文件
[root@zzuli named]# pwd
/var/named/chroot/var/named 所放置的文件夹
[root@zzuli named]#
```

这两个文件中的内容如下图所示：

```
root@zzuli:/var/named/chroot/var/named
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

STTL 600 :这个跟清楚cache的时间有关！单位是秒！
⑥ IN SOA localhost. root.localhost. (
2002120601 :serial 于master及slave是否同步有关
:一般而言，如果这个数值变大了，slave才会同步更新
28800 :Refresh定义出slave多久会主动的检查serial
:的值，以便主动的更新数据库
14400 :Retry 定义出，如果slave没有连上master DNS
:主机则多久后会重新再次的主动检查
720000 :Expire 如果一直没有连接上mater,那么到了这个时候slave
:就会放弃检查的动作了，不再更新

86400 ) :Minimum这个其实就是TTL啦 如果您没有定义TTL，那么
:TTL的值就以这个来设定
:开始设定正解的信息内容：
⑥ IN NS localhost. :特别留意最后面有个 .
localhost. IN A 127.0.0.1
:A是正解里面hostname对应的IP的标志

~
~
~
~
~

"named.localhost" 18L, 861C 13.4-25 全部
```

root@zzuli:/var/named/chroot/var/named

文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

```

$TTL 600
IN      SOA      www.apache.com. root.www.apache.com. (
2002120601
28800
14400
720000
86400 )
IN      NS       www.apache.com
www     IN      A       192.168.55.2

```

这个为 \$TTL

这个为 @ 符号

named.apache.com" 10L, 148C

7,2-9 全部

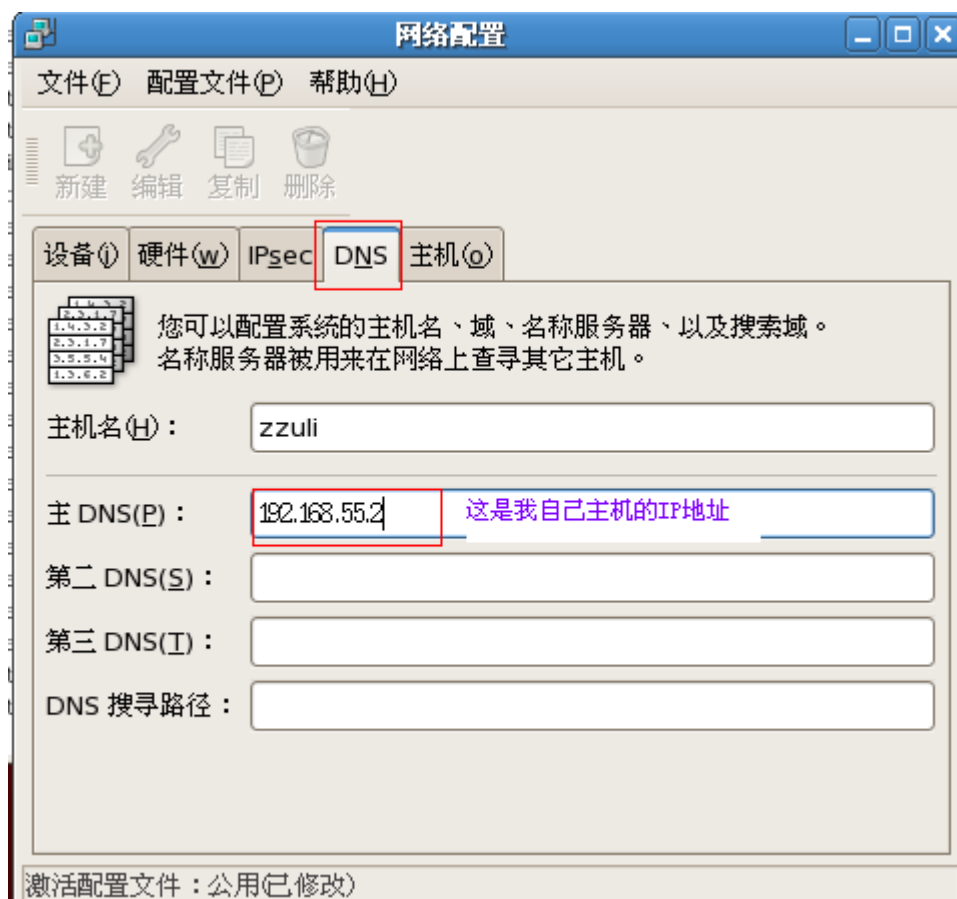
这些配置文件都写好后，就可以启动 DNS 服务了。

```
root@zzuli:/var/named/chroot/var/named
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

system-config-keyboard      system-config-time
system-config-language     system-config-users
system-config-lvm          system-control-network
system-config-network      system-install-packages
system-config-network-cmd  systool
system-config-network-gui  sys-unconfig
system-config-network-tui

[root@zzuli named]# system-config-network
[root@zzuli named]# system-config-network
[root@zzuli named]# service named stop  停止服务
停止 named:                                [确定]
[root@zzuli named]# service named start  启动服务
启动 named:                                [确定]
[root@zzuli named]# service named status  查看服务状态
number of zones: 2
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/1000
tcp clients: 0/100
server is up and running
[root@zzuli named]#
```

然后你需要把自己的 DNS 设置成自己的 IP 地址。



现面进行测试，还机的我们在将 Apache 服务器的时候做的那个小网页吗？我们以前是通过 IP 地址来访问的，现在我们来通过域名来访问吧。192.168.55.2 在我的配置中所对应的域名为 www.apache.com 下面来让我们验证一下：



欢迎学习 **Apache** 服务器

我们还可一通过 nslookup 命令来测试:

```
[root@zzuli named]# nslookup www.apache.com 使用nslookup命令
Server:          192.168.55.2 服务器地址
Address:         192.168.55.2#53 服务器加端口地址大家知道dns的端口为53

Name:   www.apache.com 域名
Address: 192.168.55.2 域名所对应的IP地址

[root@zzuli named]# █
```

Samba

在介绍本实验前,先说明一下我用到的两个机器:一台 linux 机器 IP 为 192.168.55.2
一台是 windows xp 它的 IP 是 192.168.55.1

下面我们来介绍一下基本的 samba 服务器的简单应用。

首先我们来说明一下, samba 服务器的配置文件是 smb.conf 这个文件在/etc/samba 目录下。还有一个是 lmhosts 这个文件是 NetBIOS 名称解析(一边不用配置也可)。这里我配置了一下:

```
[root@zzuli ~]# cd /etc/samba/      samba配置文件的目录
[root@zzuli samba]# ls
lmhosts  smb.conf  smbusers  两个主要的配置文件
[root@zzuli samba]# vim lmhosts
[root@zzuli samba]#
```



```
root@zzuli:/etc/samba
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
127.0.0.1 localhost      这个就是lmhosts文件里我写的名称
192.168.55.2 linux      对应表
192.168.55.1 windows    192.168.55.2 是一台linux主机
                        192.168.55.1 是一台windows主机
```

在配置 smb.conf 配置文件之前我建议大家先这个文件给备份一下:

```
[root@zzuli samba]# cp smb.conf smb.conf.bak 备份文件
[root@zzuli samba]# ls
lmhosts  smb.conf  smb.conf.bak  smbusers
[root@zzuli samba]#
```

我建议大家不要直接用 smb.conf 里的内容,可以把它里面的内容全部删除后,自己从新写:

```
[root@zzuli samba]# echo >smb.conf 删除smb.conf里面的
[root@zzuli samba]# vim smb.conf 内容
[root@zzuli samba]#
```

下图是我写的 smb.conf 的内容:

```
root@zzuli:/etc/samba
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

[global]
workgroup=endless
netbios name=linux
server string=testing SAMBA Server
log file=/var/log/samba/log.%m
max log size=0 直接用于共享不需要设置密码，当然这样
security=share 是不安全的
socket options=TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
interfaces=192.168.55.2
dns proxy=no

[web]
comment=Web files 这里我共享了/var/www/html目录这里主要是放置网页
path=/var/www/html 的。一会用windows访问linux的时候你看到这个目
#真正的开放出来的路径在这里 录（在windows上的名字为 web）
read only=no
public=yes
#上面两个设定是告诉大家，不但可以存取
#也可以让大家查询
~
~
~
```

下面我们可以做一下测试：

```
[root@zzuli samba]# testparm  samba的测试命令
Load smb config files from /etc/samba/smb.conf
Processing section "[web]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

```
[global]
workgroup = ENDLESS
netbios name = LINUX
server string = testing SAMBA Server
interfaces = 192.168.55.2
security = SHARE
log file = /var/log/samba/log.%m
max log size = 0
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
dns proxy = No

[web]
comment = Web files
path = /var/www/html
read only = No
guest ok = Yes
[root@zzuli samba]#
```

然后我们就可以启动 samba 服务器了：

```

[root@zzuli samba]# service smb start
启动 SMB 服务:
启动 NMB 服务:
[root@zzuli samba]# service smb stop
关闭 SMB 服务:
关闭 NMB 服务:
[root@zzuli samba]# service smb status
smbd 已停
nmbd 已停
[root@zzuli samba]#

```

samba服务器的启动, 停止, 查看状态命令 [确定]
[确定]
[确定]
[确定]

下面我们在 linux 主机上做一些简单的测试:

```

[root@zzuli samba]# smbclient -L //linux
Password: 输入你现在所用的用户密码, 我用的用户是root
Domain=[ENDLESS] OS=[Unix] Server=[Samba 3.0.23c-2]

```

Sharename	Type	Comment
web	Disk	Web files
IPC\$	IPC	IPC Service (testing SAMBA Server)

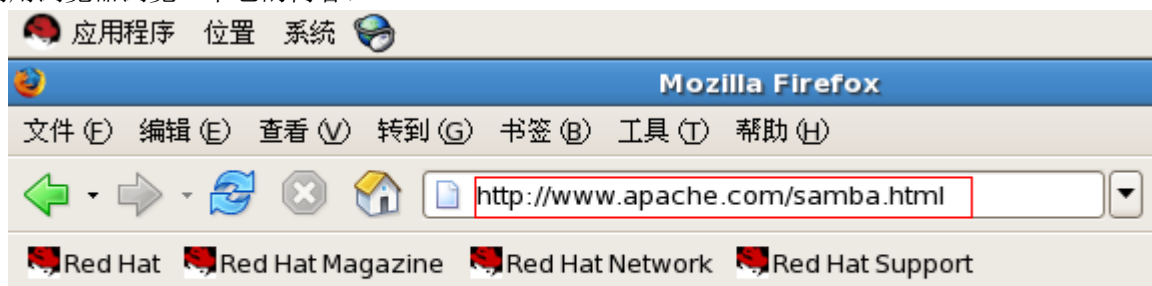
Domain=[ENDLESS] OS=[Unix] Server=[Samba 3.0.23c-2]

Server	Comment
LINUX	testing SAMBA Server
Workgroup	Master
ENDLESS	

[root@zzuli samba]#

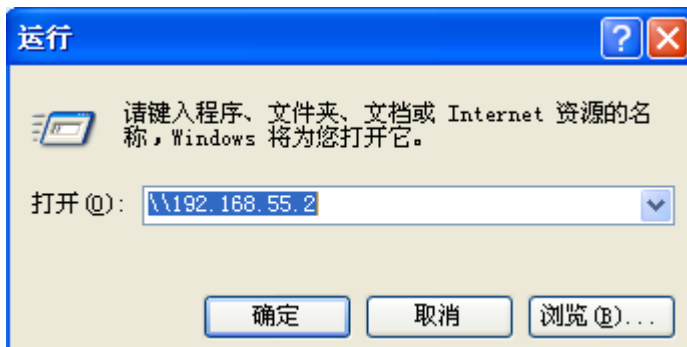
下面这些就是samba服务器现在的一些信息, 比如共享的目录web等。。。。

下面我们在共享的/var/www/html 目录下, 做一个简单的 html 文件。起名为 samba.html, 我们用浏览器浏览一下它的内容:

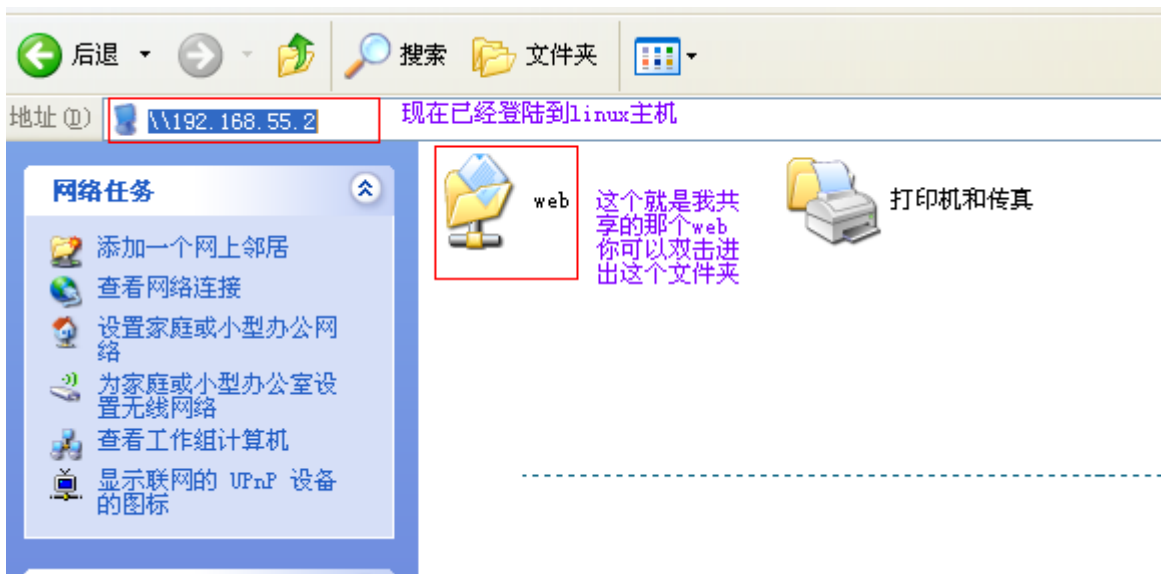


这就是网页中的内容, 就是简单的一句话

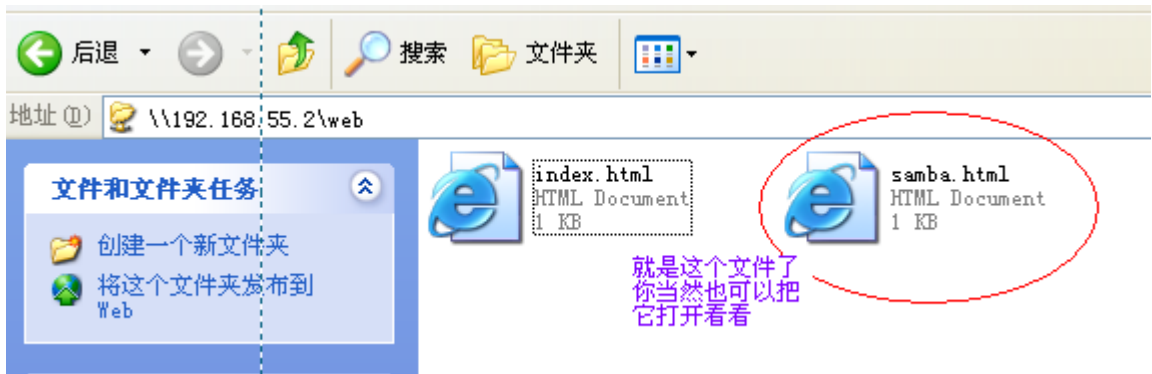
下面我们用 windows 来登陆到共享目录:
在 windows 的运行中输入 [\\192.168.55.2](http://192.168.55.2) 即可:



然后就会登陆到共享的目录中了:



进入 web 文件夹后, 就会看到 samba.html 这个 HTML 文件了:



我选择用记事本打开了这个文件, 下面是它的内容其实就一句话:



DHCP

在做实验之前我们来看看实验的环境：我需要两台机器，一台是 DHCP 服务器（Linux 主机）这里我用的是 Redhat 企业版 5。另一台是客户机：我用的是 windows xp 系统。

Linux 主机的 IP 是：192.168.55.2 。我要配置的 DHCP 服务器的地址池范围是：

192.168.55.4 — 192.168.55.10

DHCP 服务器的简单配置:

DHCP 服务器和 apache 服务器一样,它也是有一个配置文件,这个文件就在/etc 目录下,名字为 dhcpd.conf。我们要编辑这个配置文件,如果配置文件没有编辑正确的话,那么 DHCP 是无法启动的。

```
[root@zzuli ~]# vim /etc/dhcpd.conf 用VI编辑器打开DHCP的配置文件
```

我们需要把配置文件的内容写成如下样子:

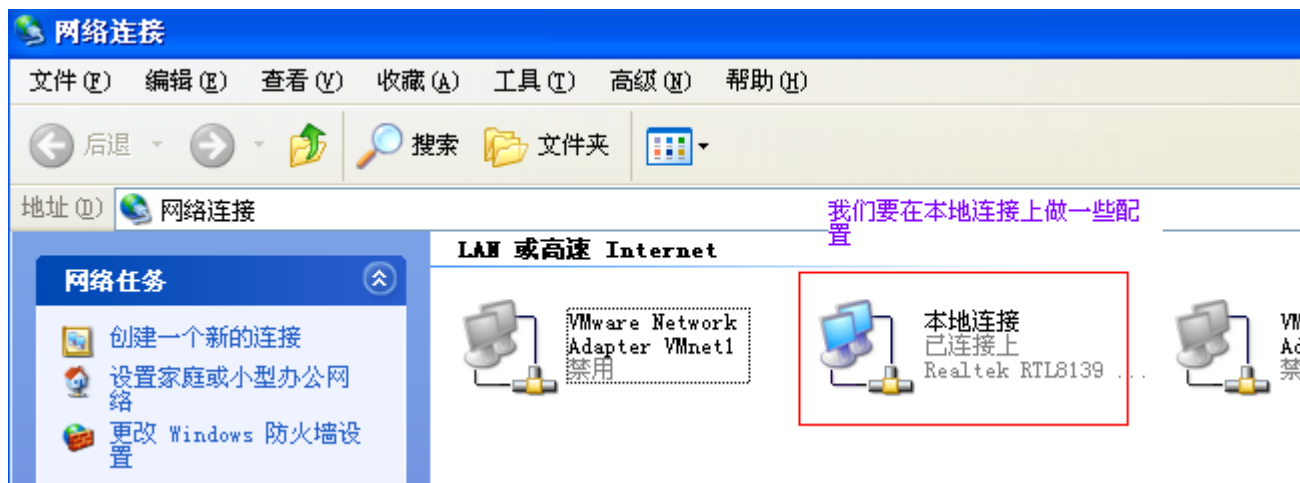
```
root@zzuli:~  
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)  
ddns-update-style interim; 这个与DHCP的启动有关  
ignore client-updates;  
  
subnet 192.168.55.0 netmask 255.255.255.0 {  
  
# --- default gateway  
    option routers                192.168.55.2; #设置网关  
    option subnet-mask            255.255.255.0;#设置子网掩码  
  
    option nis-domain             "endless.linux";  
    option domain-name           "endless.linux";  
    option domain-name-servers   192.168.0.2; #设置DNS  
    range dynamic-bootp 192.168.55.4 192.168.55.10; #设置动态IP范围  
    default-lease-time 21600; #设置IP的有效期  
    max-lease-time 43200;  
}  
~  
~  
~  
~  
~  
~
```

配置完之后就可以启动 DHCP 服务器了:

```
[root@zzuli ~]# service dhcpd stop
关闭 dhcpd:
[root@zzuli ~]# service dhcpd start
启动 dhcpd:
[root@zzuli ~]# service dhcpd status
dhcpd (pid 4116) 正在运行...
[root@zzuli ~]#
```

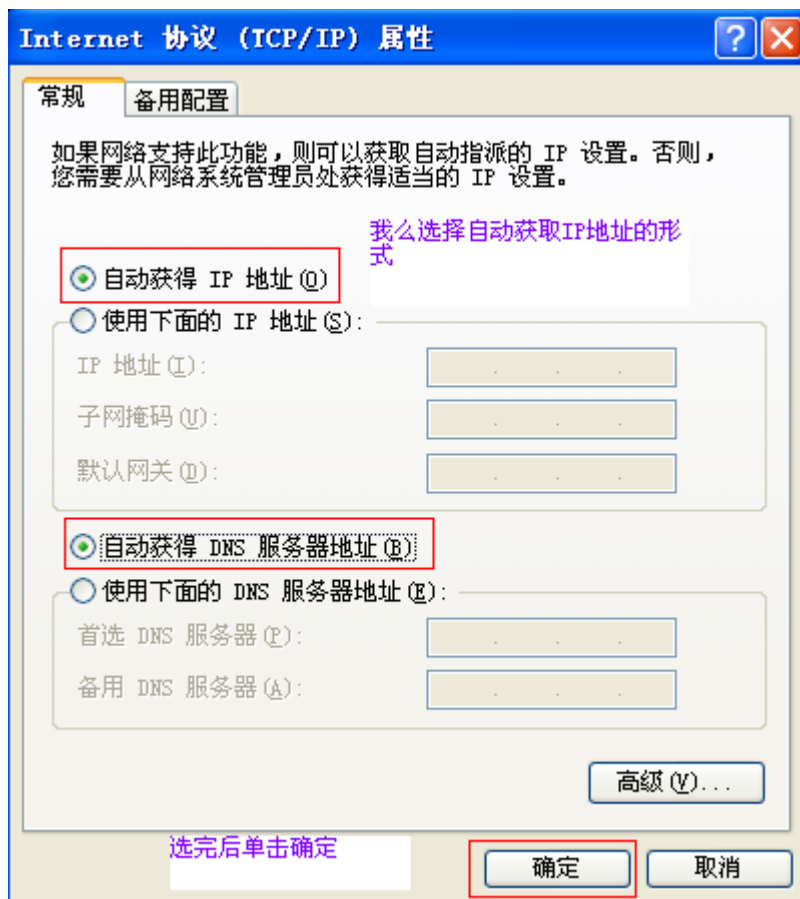
然后让我们来验证一下:

我们登陆到 windows XP 主机上，右键单击网上邻居，查看属性，会出现下图所示：



接着我们右键单击本地连接，查看属性，会出现下图：





然后我们在运行里输入“cmd”：



在终端用 ipconfig /all 命令查看 IP 地址情况：

Ethernet adapter 本地连接:

还记得我们在 dhcpd.conf 里边的配置吗

```
Connection-specific DNS Suffix . : endless linux
Description . . . . . : Realtek RTL8139 Family PCI Fast Ethernet NIC
Physical Address. . . . . : 00-E0-13-03-32-3D
Dhcp Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . : 192.168.55.9
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.55.2
DHCP Server . . . . . : 192.168.55.2
DNS Servers . . . . . : 192.168.0.2
Lease Obtained. . . . . : 2008年7月18日 12:49:04
Lease Expires . . . . . : 2008年7月18日 18:49:04
```

这说明我们用的是 DHCP 方式

这是我们获得的 IP 地址子网掩码和网关

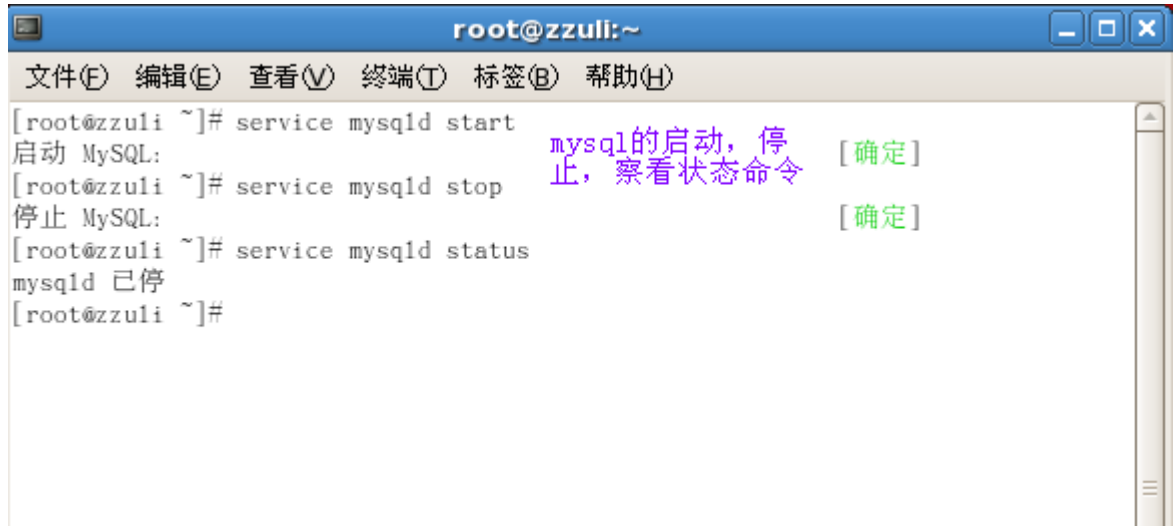
这里是我们用的 DHCP 服务器的地址，和 DNS 的地址

允许我们使用的时间限制

C:\Documents and Settings\zhou>

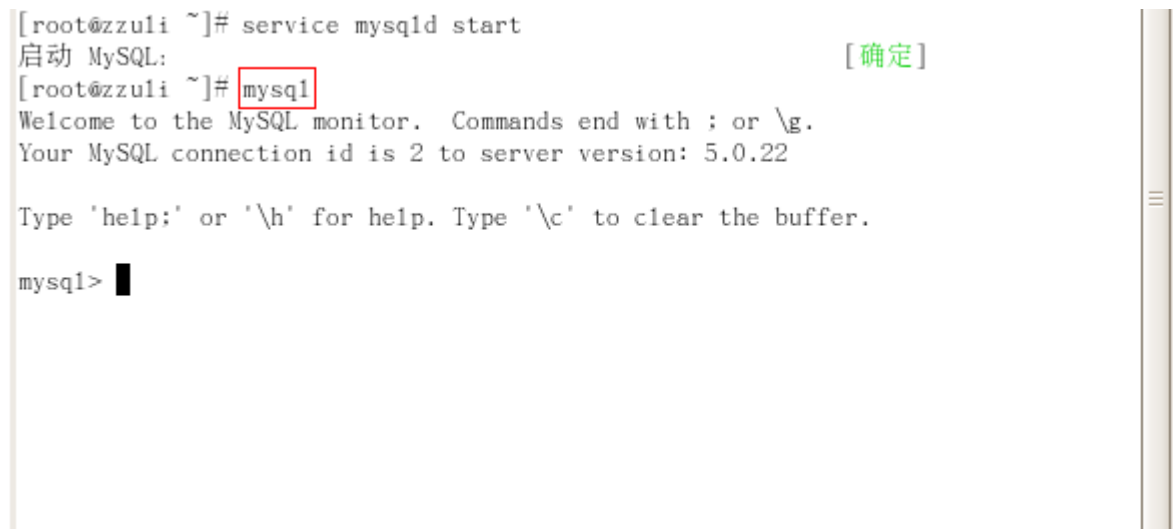
Mysql

Linux 下最常用的中小型数据库就是 mysql 了，这里我讲一下 mysql 的基本应用，因为我们的系统在安装的时候已经默认安装上了这个数据库，而且它和其它的服务不一样不需要什么配置文件，所以我们讲一下如何应用：

A terminal window titled 'root@zzuli:~' with a menu bar containing '文件(F)', '编辑(E)', '查看(V)', '终端(T)', '标签(B)', and '帮助(H)'. The terminal shows the following commands and output:

```
[root@zzuli ~]# service mysqld start
启动 MySQL:          mysql的启动，停止，察看状态命令 [确定]
[root@zzuli ~]# service mysqld stop
停止 MySQL:          [确定]
[root@zzuli ~]# service mysqld status
mysqld 已停
[root@zzuli ~]#
```

当你你启动了服务器后，你就可以用 mysql 这个命令进入命令行界面：

A terminal window showing the MySQL command-line interface. The terminal shows the following commands and output:

```
[root@zzuli ~]# service mysqld start
启动 MySQL:          [确定]
[root@zzuli ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

用这个命令进入控制台后，你就可以键入各种各样的命令了，我们这里要学会这么建立数据库，这么建立表，这么在表中添加数据，这么察看表中的数据：

```
mysql> create database zhou
-> ;
Query OK, 1 row affected (0.01 sec)

mysql> show databases:
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
| zhou |
+-----+
4 rows in set (0.12 sec)

mysql>
```

我们先用 create 命令创建了一个数据库，然后用 show 命令查看我们有多少个数据库，看看我们创建的数据库“zhou”在不在其中，然后我们可以用 use 命令选择我们需要操作的数据库：

```
mysql> use zhou;
Database changed
mysql>

mysql> create table test(id int,name varchar(20));
Query OK, 0 rows affected (0.09 sec)

mysql>
```

我们选择了我们需要的数据库后，我们用 create table 命令在这个数据库中创建一个表，表名为 test。如果在没有给表中添加任何数据的情况下，察看这个表的话，那么给你这个表示空的这样的信息，如下图所示：

```
mysql> select * from test;
Empty set (0.06 sec)

mysql>
```

看果然后这样的提示吧。

接着我们需要在这表中添加一些数据，我们用 insert into 命令：

```
mysql> insert into test values(1,'zhoubingfeng');
Query OK, 1 row affected (0.01 sec)

mysql>
```

我们在这个表中插入了一条记录，下面我们可以用 select 命令来查询表中的信息了：

```
mysql> select * from test;
+----+-----+
| id | name |
+----+-----+
| 1  | zhoubingfeng |
+----+-----+
1 row in set (0.00 sec)

mysql>
```

这个就是查询的结果。这里只是介绍了一下 `mysql` 的基本应用，其实 `mysql` 很复杂的，这里只是做一下介绍。

FTP

这里我们使用的 FTP 是 VSFTP，我们做实验的平台是 redhat 企业版 5。
其实 ftp 的配置比较简单，我们是在安装系统的时候默认安装上 VSFTP 的。
VSFTP 的配置文件在/etc/vsftp 这个目录下，名字为 vsftp.conf。



下面我们来看看 vsftp 配置文件里的内容吧：

```
listen_port=21
    使用的 vsftpd 命令通道的 port number 设定，如果您想要使用非
    正规的 ftp port，在这个设定项目修改吧！
dirmessage_enable=YES (NO)
    当使用者进入某个目录时，会显示该目录需要注意的内容，显示的
    档案预设是 .message ，当然，可以使用底下的设定项目来修订！
```

message_file=.message

当 dirmessage_enable=YES 时，可以设定这个项目来让 vsftpd 寻找该档案来显示讯息！您也可以设定其它档名喔！

listen=YES (NO)

若设定为 YES 表示 vsftpd 是以 standalone 的方式来启动的！

pasv_enable=YES (NO)

启动被动式联机(passive mode)，一定要设定为 YES 的啦！

use_localtime=YES (NO)

是否使用主机的时间？！预设使用 GMT 时间(格林威治)，会比台湾时间晚 8 小时，一般来说，建议设定为 YES 吧！

write_enable=YES (NO)

是否允许使用者具有写入的权限？！这包括删除与修改等功能喔！

connect_timeout=60

单位是秒，如果 client 尝试连接我们的 vsftpd 命令通道超过 60 秒，则不等待，强制断线咯。

accept_timeout=60

当使用者以被动式 PASV 来进行数据传输时，如果主机启用 passive port 并等待 client 超过 60 秒，那么就给他强制断线！您可以修改 60 这个数值。

data_connection_timeout=300

如果 client 与 Server 间的数据传送在 300 秒内都无法传送成功，那 Client 的联机就会被我们的 vsftpd 强制剔除！

idle_session_timeout=300

如果使用者在 300 秒内都没有命令动作，强制离线！

`max_clients=0`

如果 vsftpd 是以 stand alone 方式启动的，那么这个设定项目可以设定同一时间，最多有多少 client 可以同时连上 vsftpd 哩！？

`max_per_ip=0`

与上面 max_clients 类似，这里是同一个 IP 同一时间可允许多少联机？

`pasv_max_port=0`

`pasv_min_port=0`

上面两个是与 passive mode 使用的 port number 有关，如果您想要使用 65400 到 65410 这 11 个 port 来进行被动式资料的连接，可以这样设定 `pasv_max_port=65410` 以及 `pasv_min_port=65400`

`ftpd_banner=一些文字说明`

当使用者无法顺利连上我们的主机，例如联机数量已经超过 max_clients 的设定了，那么 client 的画面就会显示『一些文字说明』的字样，您可以修改

关于实体用户登入者的设定值

`guest_enable=YES (NO)`

若这个值设定为 YES 时，那么任何非 anonymous 登入的账号，均会被假设成为 guest (访客) 喔！

`local_enable=YES (NO)`

这个设定值必须要为 YES 时，在 /etc/passwd 内的账号才能以实体用户的方式登入我们的 vsftpd 主机喔！

`local_max_rate=0`

实体用户的传输速度限制，单位为 bytes/second， 0 为不限制。

`chroot_local_user=YES (NO)`

将使用者限制在自己的家目录之内(chroot)！这个设定在 vsftpd 当中预设是 NO，因为有底下两个设定项目的辅助喔！
所以不需要启动他！

`chroot_list_enable=YES (NO)`

是否启用将某些实体用户限制在他们的家目录内？！预设是 NO，不过，如果您想要让某些使用者无法离开他们的家目录时，可以考虑将这个设定为 YES，并且规划下个设定值

`chroot_list_file=/etc/vsftpd.chroot_list`

如果 `chroot_list_enable=YES` 那么就可以设定这个项目了！他里面可以规定那一个实体用户会被限制在自己的家目录内而无法离开！(chroot) 一行一个账号即可！

`userlist_deny=YES (NO)`

若此设定值为 YES 时，则当使用者账号被列入到某个档案时，在该档案内的使用者将无法登入 vsftpd 服务器！该档案文件名与下列设定项目有关。

`userlist_file=/etc/vsftpd.user_list`

若上面 `userlist_deny=YES` 时，则这个档案就有用处了！在这个档案内的账号都无法使用 vsftpd 喔！

关于匿名者登入的设定值

anonymous_enable=YES (NO)

设定为允许 anonymous 登入我们的 vsftpd 主机！预设是 YES，底下的所有相关设定都需要将这个设定为 anonymous_enable=YES 之后才会生效！

anon_world_readable_only=YES (NO)

仅允许 anonymous 具有下载可读档案的权限，预设是 YES。

anon_other_write_enable=YES (NO)

是否允许 anonymous 具有写入的权限？预设是 NO！如果要设定为 YES，那么开放给 anonymous 写入的目录亦需要调整权限，让 vsftpd 的 PID 拥有者可以写入才行！

anon_mkdir_write_enable=YES (NO)

是否让 anonymous 具有建立目录的权限？默认值是 NO！如果要设定为 YES，那么 anon_other_write_enable 必须设定为 YES！

anon_upload_enable=YES (NO)

是否让 anonymous 具有上传数据的功能，预设是 NO，如果要设定为 YES，则 anon_other_write_enable=YES 必须设定。

deny_email_enable=YES (NO)

将某些特殊的 email address 抵挡住，不让那些 anonymous 登入！
如果以 anonymous 登入主机时，不是会要求输入密码吗？密码不是要您输入您的 email address 吗？如果你很讨厌某些 email address，就可以使用这个设定来将他取消登入的权限！需与下个设定项目配合：

banned_email_file=/etc/vsftpd.banned_emails

如果 deny_email_enable=YES 时，可以利用这个设定项目来规定那个

捕捉指定区域 Win + < 4 >

上面就是 vsftp 配置文件里的内容了。

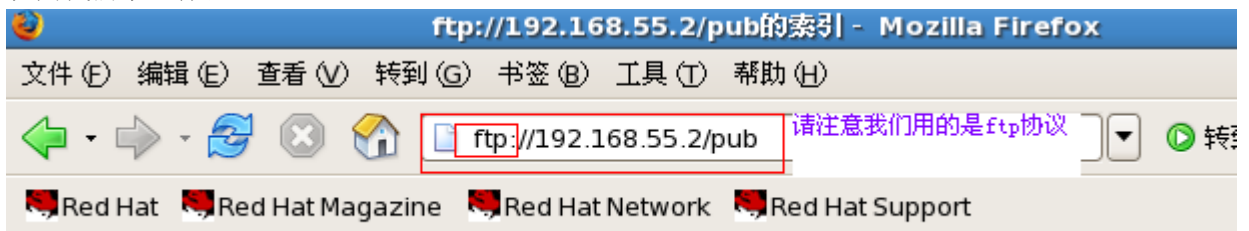
下面我们来看看启动 vsftp 的基本命令，这和上面提到过的其他服务器是一样的：

```
[root@zzuli ~]# service vsftpd start    vsftp启动，停止，查看服务器状态的命令 [确定]
为 vsftpd 启动 vsftpd:
[root@zzuli ~]# service vsftpd stop      [确定]
关闭 vsftpd:
[root@zzuli ~]# service vsftpd status
vsftpd 已停
[root@zzuli ~]#
```

请注意： vsftp 的默认的目录是在/var/vsftp/pub 这个目录下的：

```
[root@zzuli ~]# cd /var/ftp/pub/    vsftp目录的默认路径
[root@zzuli pub]# ls
[root@zzuli pub]# pwd
/var/ftp/pub
[root@zzuli pub]# mkdir myftp    我们在这个目录下创建一个文件夹
[root@zzuli pub]# ls            , myftp, 用来一会验证的时候用
myftp
[root@zzuli pub]#
```

现面我们来进行验证:



ftp://192.168.55.2/pub的索引

[回到上一层文件夹](#)

这个就是我们刚才建立的文件夹

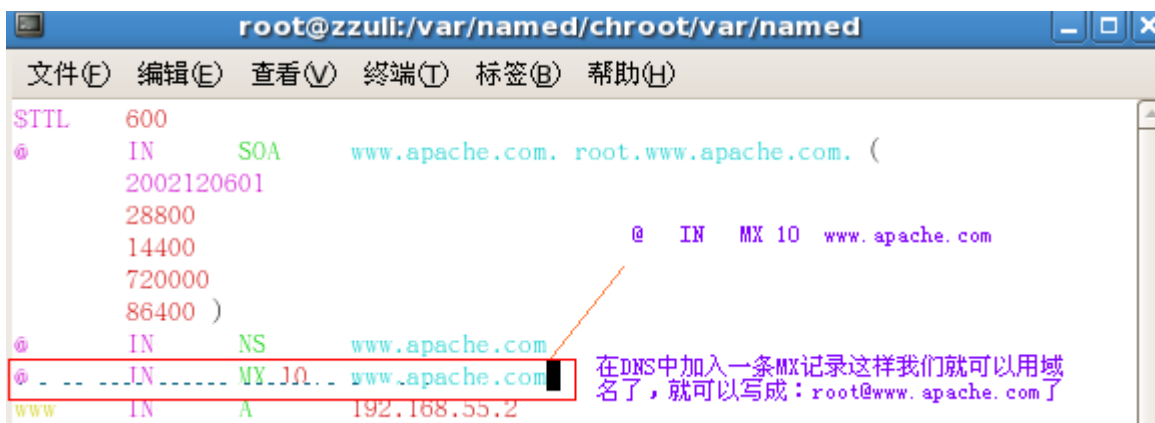
 [myftp](#)

2008年07月22日 02:00:00

Sendmail

下面我们来介绍一下 linux 下的邮件服务器 Sendmail 的架设过程，这里我们只介绍基础架设。在我们用邮件服务器的时候，我们同时也会用到 DNS 服务器，我们要在 DNS 服务器上配置 MX 记录，这样我们就可以用例如：zhou@zzuli.edu.cn 这样的域名了。这样也十分的方便我们记住这些名字。

下面我们先来设置一下 DNS，主要就是在 DNS 中加入一个 MX 记录：



```
root@zzuli:/var/named/chroot/var/named
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

$TTL      600
@         IN      SOA     www.apache.com. root.www.apache.com. (
2002120601
28800
14400
720000
86400 )
@         IN      NS      www.apache.com
@         IN      MX      10 www.apache.com
www       IN      A       192.168.55.2
```

在DNS中加入一条MX记录这样我们就可以用域名了，就可以写成：root@www.apache.com了

好的，现在我们已经设置完了 DNS。设置完后要重新启动 DNS 服务器。那么下面我们来介绍一下 Sendmail 邮件服务器。

首先，我们要看看自己的机器上是否安装的有 Sendmail 的组建，我们要用 rpm 命令查看，如果没有的话，那么我们就需要自己安装了。



```
root@zzuli:/home/Server
文件(E) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)

[root@zzuli Server]# rpm -qa | grep sendmail
sendmail-cf-8.13.8-2.e15
sendmail-8.13.8-2.e15
[root@zzuli Server]# rpm -qa | grep m4
m4-1.4.5-3.e15.1
[root@zzuli Server]# rpm -q mailx
mailx-8.1.1-44.2.2
[root@zzuli Server]# rpm -qa | grep imap
cyrus-imapd-per1-2.3.7-1.1.e15
cyrus-imapd-utils-2.3.7-1.1.e15
cyrus-imapd-2.3.7-1.1.e15
cyrus-imapd-devel-2.3.7-1.1.e15
[root@zzuli Server]#
```

查看各个组建是否全部都安装上去了。如果没有全部安装上去的话，那么请用rpm方式安装。sendmail的tar安装方式十分的复杂。

好，我用的系统是 redhat 企业版 5，Sendmail 必须的组件我都已经安装好了。

Sendmail 的配置文件在/etc/mail 底下，我们一般不手动修改它的配置文件，而是用 m4 程序来生成和修改它。这个配置文件的名字为：sendmail.cf



```
root@zzuli:/etc/mail
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@zzuli mail]# cd /etc/mail
[root@zzuli mail]# m4 /usr/share/sendmail-cf/m4/cf.m4 \
> sendmail.mc >redhat.cf
[root@zzuli mail]# vim sendmail.mc
[root@zzuli mail]# vim redhat.cf
[root@zzuli mail]# m4 sendmail.mc >redhat.cf
[root@zzuli mail]#
```

用m4程序配置sendmail的主要配置文件。m4程序是主要配置sendmail配置文件的程序。

```
root@zzuli:/etc/mail
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@zzuli mail]# cp redhat.cf /etc/mail/sendmail.cf
cp: 是否覆盖 /etc/mail/sendmail.cf? y
[root@zzuli mail]#
```

替换掉默认的配置文件的sendmail.cf，用我们用m4程序生成的redhat.cf来替换掉它

像这样生成的配置文件只能接收 127.0.0.1 的信件，如果我们需要接收其它的信件的话，那么我们需要稍微修改一下这个配置文件：



```
root@zzuli:/etc/mail
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
FEATURE(`blacklist_recipients')dn1
EXPOSED_USER(`root')dn1
dn1 #
dn1 # For using Cyrus-IMAPd as POP3/IMAP server through LMTP delivery uncomment
dn1 # the following 2 definitions and activate below in the MAILER section the
dn1 # cyrusv2 mailer.
dn1 #
dn1 define(`confLOCAL_MAILER', `cyrusv2')dn1
dn1 define(`CYRUSV2_MAILER_ARGS', `FILE /var/lib/imap/socket/lmtp')dn1
dn1 #
dn1 # The following causes sendmail to only listen on the IPv4 loopback address
dn1 # 127.0.0.1 and not on any other network devices. Remove the loopback
dn1 # address restriction to accept email from the internet or intranet.
dn1 #
dn1 DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dn1
dn1 #
dn1 # The following causes sendmail to additionally listen to port 587 for
dn1 # mail from MUAs that authenticate. Roaming users who can't reach their
dn1 # preferred sendmail daemon due to port 25 being blocked or redirected find
dn1 # this useful.
dn1 #
dn1 DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dn1
dn1 #
```

这句话需要修改把127.0.0.1改成0.0.0.0

捕捉指定区域 Win 116,39 66%

```
root@zzuli:/etc/mail
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
FEATURE(`blacklist_recipients')dn1
EXPOSED_USER(`root')dn1
dn1 #
dn1 # For using Cyrus-IMAPd as POP3/IMAP server through LMTP delivery uncomment
dn1 # the following 2 definitions and activate below in the MAILER section the
dn1 # cyrusv2 mailer.
dn1 #
dn1 define(`confLOCAL_MAILER', `cyrusv2')dn1
dn1 define(`CYRUSV2_MAILER_ARGS', `FILE /var/lib/imap/socket/lmtp')dn1
dn1 #
dn1 # The following causes sendmail to only listen on the IPv4 loopback address
dn1 # 127.0.0.1 and not on any other network devices. Remove the loopback
dn1 # address restriction to accept email from the internet or intranet.
dn1 #
dn1 # 这是更改后的内容
DAEMON_OPTIONS(`Port=smtp,Addr=0.0.0.0, Name=MTA')dn1
dn1 #
dn1 # The following causes sendmail to additionally listen to port 587 for
dn1 # mail from MUAs that authenticate. Roaming users who can't reach their
dn1 # preferred sendmail daemon due to port 25 being blocked or redirected find
dn1 # this useful.
dn1 #
dn1 DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dn1
dn1 #
"sendmail.mc" 176L, 7207C 已写入 116,40 66%
```

然后，我们需要重新生成配置文件：

```
root@zzuli:/etc/mail
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@zzuli mail]# vim sendmail.mc
[root@zzuli mail]# m4 sendmail.mc >redhat.cf 重新生成配置文件
[root@zzuli mail]# mv sendmail.cf sendmail.cf.old 备份配置文件
[root@zzuli mail]# cp redhat.cf sendmail.cf 替换掉原来的配置文件
[root@zzuli mail]#
```

现在我们可以启动 Sendmail 服务器了：

```
[root@zzuli mail]# service sendmail start
启动 sendmail:
启动 sm-client:
sendmail (pid 2879 2871) 正在运行...
[root@zzuli mail]# service sendmail status
[root@zzuli mail]# service sendmail stop
关闭 sm-client:
关闭 sendmail:
[root@zzuli mail]#
```

sendmail的启动，
停止和查看状态的
命令。
[确定]
[确定]
[确定]
[确定]
[确定]

下面我们来验证：

这里我用了两个用户来进行验证，一个是 root 用户，一个是新建的 test 用户。他们两个邮件放置在/var/mail 目录下：

```
[root@zzuli named]# mail root@www.apache.com 写信命令
Subject: This is a test 信的title
Hello sendmail!
. ["."]为退出命令
Cc: quit

Date: Thu, 24 Jul 2008 11:16:46 +0800
From: root <root@localhost.localdomain>
Message-Id: <200807240316.m603Gkt6003120@localhost.localdomain>
To: root@www.apache.com 查看root用户的邮件，这个root用户的邮件
Subject: This is a test 在"/var/mail/root"内
Cc: quit@localhost.localdomain

Hello sendmail!

--m603HQj3003123.1216869446/localhost.localdomain--
```

297,15 底端

下面我们使用 test 用户验证：

```
[root@zzuli /]# useradd test 新建一测试用户，它的邮件在
[root@zzuli /]# passwd test "/var/mail/test" 中
Changing password for user test.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@zzuli /]# █

[test@zzuli /]$ mail test@www.apache.com
Subject: this is a test
this test is for user test!
Cc: quit
[test@zzuli /]$ /home/test/dead.letter... Saved message in /home/test/dead.lette
r
█

From: test@localhost.localdomain
Message-Id: <200807240324.m6030UCS003191@localhost.localdomain>
To: test@www.apache.com 查看test用户的邮件
Subject: this is a test
Cc: quit@localhost.localdomain

this test is for user test!

--m603PArF003194.1216869910/localhost.localdomain--
```

64,1 底端