

1. nfs自动挂载技术:

autofs服务程序与mount命令不同之处在于它是一种**守护进程**，只有检测到用户试图访问一个尚未挂载的文件系统时才自动的检测并挂载该文件系统。

autofs非常方便，主要有两点:

- 1、设置开机不一定要挂载的目录，当用的时候才实现自动挂载。
- 2、用户不使用自动挂载的目录一段时间，会自动卸载。（默认时间为5分钟），可以在autofs.conf

1.1 安装autofs服务:

```
# 在客户端执行以下命令:  
yum install autofs -y
```

1.2 编辑/etc/auto.master:

```
# 添加以下行:  
/mnt/nfs /etc/nfs.misc
```

/mnt/nfs是总的访问目录(客户端的目录) /etc/nfs.misc是对总访问目录的描述，用于子目录的编辑，用户权限分离

1.3 编辑nfs.misc:

```
xiaoming -fstype=nfs,rw,sync  
192.168.75.123:/data/jfedu  
xiaowang -fstype=nfs,ro,sync  
192.168.75.123:/data/jfedu
```

1.4 启动autofs服务:

```
systemctl start autofs
```

1.5 验证权限:

```
## 登录不同目录验证权限:
```

```

[root@node4 nfs]# ls
[root@node4 nfs]#
[root@node4 nfs]#
[root@node4 nfs]# cd xiaoming
[root@node4 xiaoming]# df
Filesystem                1K-blocks    Used Available
Use% Mounted on
/dev/sda3                  19923968 1986528 17937440
10% /
devtmpfs                   231980      0    231980
0% /dev
tmpfs                      241808      0    241808
0% /dev/shm
tmpfs                      241808     4636    237172
2% /run
tmpfs                      241808      0    241808
0% /sys/fs/cgroup
/dev/sda1                  508580    114484    394096
23% /boot
tmpfs                      48364      0     48364
0% /run/user/0
192.168.75.123:/data/jfedu 20128768 2021376 18107392
11% /mnt/nfs/xiaoming

```

2. ftp文件服务器简介:

vsftpd 是“very secure FTP daemon”的缩写，是一个完全免费的、开放源代码的ftp服务器软件。特点是：非常高的安全性需求、带宽限制、良好的可伸缩性等。

2.1 工作原理:

vsftpd使用ftp协议，该协议属于应用层协议。它是典型的c/s架构，ftp服务端用来存储文件，ftp客户端可以通过ftp协议连接服务端实现上传和下载资源。

ftp使用tcp的21端口进行**命令传输**，然后用tcp的20端口进行**数据传输**（主动模式）。默认是**被动模式**。

2.2 安装部署:

```
yum install vsftpd ftp lftp -y
```

vsftpd: 为服务端软件

ftp、lftp: 为客户端工具, 推荐使用lftp

2.3 启动服务:

```
systemctl start vsftpd
```

2.4 匿名用户访问:

用ftp客户端匿名登录需要输入**用户名及密码**验证, 匿名用户名为: ftp或者anonymous, 密码为空。

用lftp客户端匿名登录则不需要输入以上信息。

ftp客户端连接:

```
[root@www ~]# ftp 192.168.0.103
Connected to 192.168.0.103 (192.168.0.103).
220 (vsFTPd 3.0.2)
Name (192.168.0.103:root): ftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

lftp客户端连接:

```
[root@www ~]# lftp 192.168.0.103
lftp 192.168.0.103:~>
```

这里我们**重点讲lftp使用方法**, ftp客户端工具使用方法大致相同。

下载命令:

2.4.1 get下载单个文件:

可以先切换到**本地指定目录** (data) 进行文件的下载, 保存:

```
lftp 192.168.0.103:/> lcd /data/
lcd 成功, 本地目录=/data
lftp 192.168.0.103:/> get file003.txt

# 下载到指定目录(默认使用源文件名, 可以自定义名):
lftp 192.168.0.103:/> get file003.txt -o /tmp
```

ps:当客户端已经连接上服务端, cd是用于切换服务器中的目录命令, 如果想切换客户端本地的目录则使用lcd命令。

2.4.2 mget批量下载:

```
lftp 192.168.0.103:/> mget file00*
Total 7 files transferred
```

默认配置只能进行文件的读取和下载, 不能进行写入和上传文件:

```
lftp 192.168.0.103:~> put /etc/fstab
put: Access failed: 550 Permission denied. (fstab)
lftp 192.168.0.103:/> mkdir abc
mkdir: Access failed: 550 Permission denied. (abc)
```

可以看到上传命令和创建命令都失败了, 没有相应的权限!

开启匿名用户创建文件, 重命名, 删除, 上传权限:

```
#开启上传权限
anon_upload_enable=YES
#开启创建文件权限
anon_mkdir_write_enable=YES
#开启重命名, 删除权限
anon_other_write_enable=YES
anon_umask=022
```

重启服务, 再次进入, 发现还是没法创建目录, 但是报错信息不一样, 如下:

```
lftp 192.168.0.103:/> mkdir abc
mkdir: Access failed: 550 Create directory operation
failed. (abc)
```

这是因为目录没有写权限，给pub目录授权，如下：

```
[root@localhost ~]# chmod o+w /var/ftp/pub/  
[root@localhost ~]# ll /var/ftp/pub/ -d  
drwxr-xrwx 3 root root 192 8月 7 08:37 /var/ftp/pub/
```

再次进入，创建目录，如下：

```
lftp 192.168.0.103:/pub> mkdir abc  
mkdir 成功，建立 `abc'
```

小总结：要想匿名用户有写的权限，一是需要服务端配置文件开启写的权限，二是所在的目录本身有其他用户写的权限！

上传命令：

2.4.3 put上传单个文件：

要想使用上传命令，需要开启上传权限和可写权限。

语法：put [OPTS] <lfile> [-o <rfile>]

一、直接上传不改名，可以省去-o rfile，如果不知道本地目录有哪些文件，可以使用!dir查看，如下：

```
lftp 192.168.0.103:/pub> !dir  
anaconda-ks.cfg  favicon.png  
lftp 192.168.0.103:/pub> put /etc/fstab  
501 bytes transferred
```

上面这个例子中，我们将本地文件fstab上传到了匿名用户的pub目录中。

二、上传文件之后，想改名，就要带-o refile了，如下

```
lftp 192.168.0.103:/pub> put /etc/fstab -o fs
501 bytes transferred
lftp 192.168.0.103:/pub> ls
drwx-----  2 14      50                6 Aug 08 06:27
abc
-rw-----  1 14      50                501 Aug 08 06:34 fs
-rw-----  1 14      50                501 Aug 08 06:34
fstab
```

上面这个例子中，我们将本地文件**fstab**上传到了匿名用户的**pub**目录并修改了文件名。

2.4.4 mput批量上传:

三、多文件上传

上传多个文件，可以使用**put**和**mput**命令上传，多个文件之间用空格分隔，如果想使用通配符，只有**mput**命令支持：

```
lftp 192.168.0.103:/pub> put /etc/fstab /etc/favicon.png
1580 bytes transferred
Total 2 files transferred
lftp 192.168.0.103:/pub> ls
drwx-----  2 14      50                6 Aug 08 06:27
abc
-rw-----  1 14      50                1079 Aug 08 06:40
favicon.png
-rw-----  1 14      50                501 Aug 08 06:40
fstab
```

```
lftp 192.168.0.103:/pub> mput /etc/f*
1688 bytes transferred
Total 4 files transferred
lftp 192.168.0.103:/pub> ls
drwx-----  2 14      50                6 Aug 08 06:27
abc
-rw-----  1 14      50                1079 Aug 08 06:41
favicon.png
```

```
-rw----- 1 14      50          70 Aug 08 06:41
filesystems
-rw----- 1 14      50         501 Aug 08 06:41
fstab
-rw----- 1 14      50          38 Aug 08 06:41
fuse.conf
```

2.4.5 下载目录:

下载远程服务器pub目录下的lutixia目录到本地root目录下:

```
lftp 192.168.0.130:/pub> lcd
lcd 成功, 本地目录=/root
lftp 192.168.0.103:/pub> mirror -c lutixia
-c      支持断点续传
```

2.4.6 上传目录:

```
lftp 192.168.0.130:/pub> mirror -cR /tmp
lftp 192.168.0.130:/pub> ls
drwx----- 2 14      50         132 Dec 05 02:58
jfedu
# 默认上传目录后, 是不能查看目录内容的, 需要添加mask权限:
vim /etc/vsftpd/vsftpd.conf
anon_umask=022
# 重新上传一个目录:
lftp 192.168.0.130:/pub> ls
drwxr-xr-x  2 14      50         132 Dec 05 03:14
lutixia2
```

2.4.7 同步远程目录:

```
# 同步当前目录文件：
lftp 192.168.0.130:/pub> lcd
lcd 成功，本地目录=/root
lftp 192.168.0.130:/pub> mirror -ce
-e 该参数会将远程目录pub下的文件同步到本地的root目录下，并且删除本地多余的文件（远程目录没有的文件）

# 同步指定目录，如果是想同步整个目录，需要在本地端路径加上目录名：
lftp 192.168.0.130:/pub> mirror -ce lutixia /root/lutixia
#同步指定目录文件，如果是同步指定目录下的文件，不同步目录本身，就不需要指目录了：
lftp 192.168.0.130:/pub> mirror -ce lutixia /root

# lutixia为远程服务器pub目录下的一个目录
# root为本地的目录
```

2.4.8 删除目录：

```
# 删除目录：
lftp 192.168.0.103:/pub> rm -rf abc

# 删除空目录：
lftp 192.168.0.103:/pub> rmdir abc
rmdir ok, `abc' removed
```

2.4.9 删除文件：

```
lftp 192.168.0.103:/pub> rm file1
rm ok, `file1' removed
```

2.5 本地用户访问：

如果使用本地用户访问，需要服务端有对应的用户存在才行。

服务端设置用户名及密码：

```
[root@localhost ~]# id lutixia
uid=1000(lutixia) gid=1000(lutixia) 组=1000(lutixia)
[root@localhost ~]# echo "lutixia" |passwd --stdin lutixia
更改用户 lutixia 的密码 。
passwd: 所有的身份验证令牌已经成功更新。
```


修改配置文件，可以设置不让匿名用户登录，只能本地用户登录：

```
vim /etc/vsftpd/vsftpd.conf

anonymous_enable=NO
local_enable=YES
```

重启服务。然后再次访问：

```
systemctl restart vsftpd

[root@www ~]# lftp 192.168.0.103
lftp 192.168.0.103:~> ls
`ls' at 0 [正在发送命令...]
```

上面这个登录，表示匿名用户已经无法登录了。

```
[root@www ~]# lftp lutixia:lutixia@192.168.0.103
lftp lutixia@192.168.0.103:~> ls
lftp lutixia@192.168.0.103:~>
```

然后用本地用户登录，就ok了。

上传和下载和匿名用户一样的使用方法。

2.5.1 限制系统用户越狱：

在安装vsftpd后不做配置的话，系统用户是可以向上切换到其他目录的。

在配置文件中，添加如下选项：

```
vim /etc/vsftpd/vsftpd.conf

chroot_local_user=YES
chroot_list_enable=NO
allow_writeable_chroot=YES

# chroot_local_user: 是否将所有用户限制在主目录，YES为启用，NO禁用。(该项默认值是NO)
# chroot_list_enable: 是否启动限制用户（特例）的名单 YES为启用，NO禁用(包括注释掉也为禁用)
```

综合理解:

| | chroot_local_user=YES | chroot_local_user=NO |
|------------------------|---|---|
| chroot_list_enable=YES | 1、所有用户都被限制在其主目录下; 2、使用chroot_list指定的用户列表, 这些用户为“例外”, 不受限制。 | 1、所有用户不被限制其主目录下; 2、使用chroot_list指定得用户列表, 这些用户作为“例外”, 受到限制。 |
| chroot_list_enable=NO | 1、所有用户都被限制在其主目录下; 2、不使用chroot_list指定的用户列表, 没有任何“例外”。 | 1、所有用户都不被限制在其主目录下; 2、不使用chroot_list指定的用户列表, 没有任何“例外”。 |

- 如果想全部限制, 所有用户都不能切换家目录, 就可以像上面的配置:

```
chroot_local_user=YES  
chroot_list_enable=NO  
allow_writeable_chroot=YES
```

- 如果想让部分用户有切换的家目录的权限, 则需要开启限制:

```
chroot_local_user=YES  
chroot_list_enable=YES  
chroot_list_file=/etc/vsftpd/chroot_list
```

chroot_list中写上要放行用户

2.6 故障解决:

如果重启或者登陆服务器时, 报错如下:

重启报错:

```
[root@localhost ~]# systemctl restart vsftpd
Job for vsftpd.service failed because the control process
exited with error code. See "systemctl status
vsftpd.service" and "journalctl -xe" for details.
```

```
[root@localhost ~]# journalctl -xe
-- Unit vsftpd.service has begun starting up.
8月 08 02:59:14 localhost.localdomain vsftpd[12751]: 500
OOPS: bad bool value in config file for: anonymous_en
8月 08 02:59:14 localhost.localdomain systemd[1]:
vsftpd.service: control process exited, code=exited
status=2
```

解决办法：这种问题一般就是空格导致的，是每一行配置后面都不能有空格，也不能跟注释。