FTP（文件传输协议）

简介

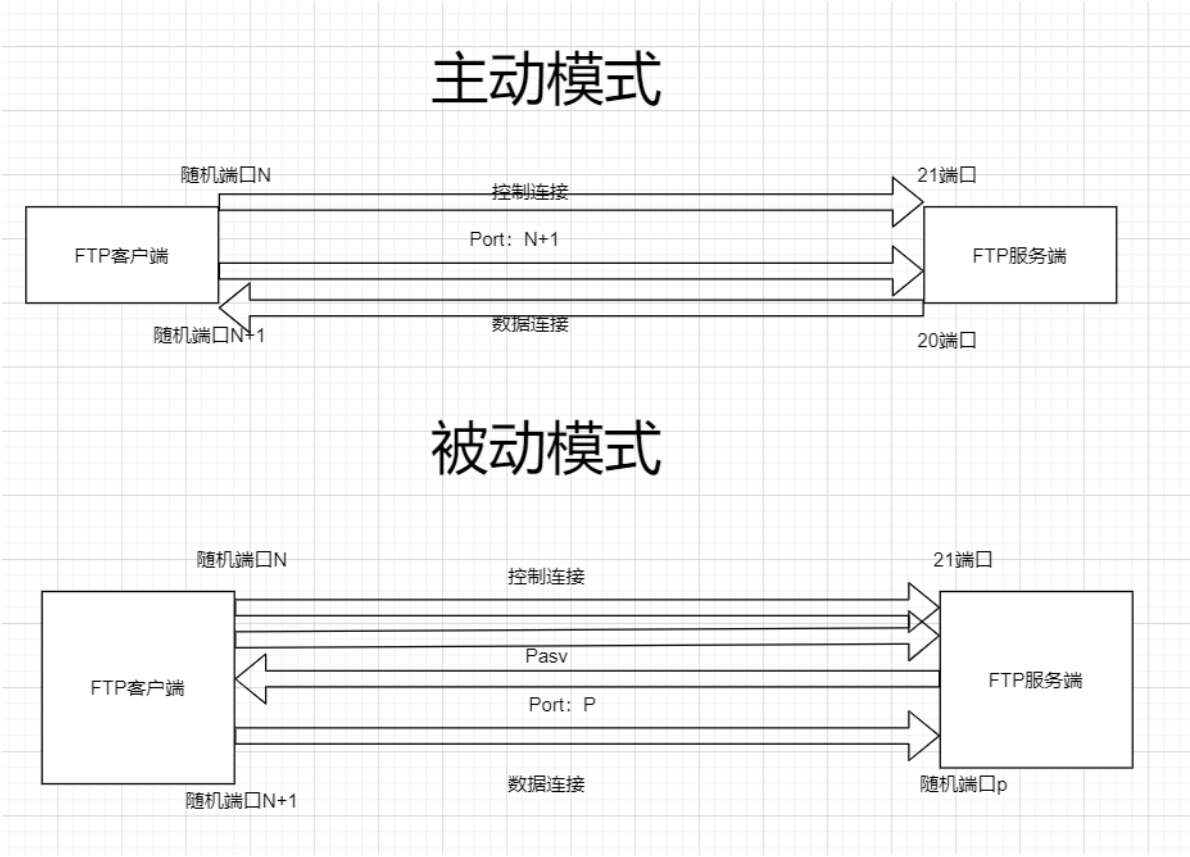
- FTP协议：文件传输协议（File Transfer Protocol）
  - 协议定义了一个在远程计算机系统和本地计算机系统之间传输文件的一个标准
  - FTP运行在OSI模型的应用层，并利用传输协议TCP在不同的主机之间提供可靠的数据传输
  - FTP 在文件传输中还支持断点续传功能，可以大幅度减少CPU网络带宽的开销
- FTP模型
  - 用户接口：提供一个用户接口并使用客户端协议解释器的服务
  - 客户端协议解释器：向远程服务器发送命令 并建立客户数据传输过程
  - 服务端协议解释器：响应客户协议机发出的命令并驱动服务端数据传输过程
  - 客户端数据传输协议：负责完成和服务器数据传输过程及客户端本地文件系统的通信
  - 服务端数据传输协议：负责完成和客户数据过程及服务器端文件系统的通信
- 控制连接（端口号21）
  - 主要用来传送在实际通信过程中需要执行的FTP命令以及命令的响应
  - 只需要很小的网络带宽
  - FTP服务端监听21端口号来等待控制连接建立
  - 建立控制连接后，还需要**验证客户身份，决定是否建立数据连接**
  - 当需要目录列表，传输文件时，才建立数据连接，并且每次客户端都是用不同的端口号来建立数据连接。数据 传输完毕，就中断这条临时的数据连接
  - 在FTP连接期间，**控制连接始终保持通常的连接状态。在数据连接存在期间，控制连接必须存在；一旦控制连 接断开，数据连接会自动关闭。**
- 数据连接（端口号20）
  - FTP服务端监听20端口来等待数据连接
  - 数据连接依赖于控制连接
  - 建立方式
- 参考博客：https://blog.csdn.net/ludan_xia/article/details/105705473
  - 主动模式
    - 通过三次握手，建立控制连接；客户端的源端口是高位随机端口，目标端口是21端口
    - 控制连接建立后，客户端进行份验证，协商数据连接采用主动模式；随后客户端会向FTP服务器发送Port报 文，表明自己监听的IP+端口，并等待FTP服务器（20端口）向自己监听的IP+端口发起数据连接请求。
    - 服务端发起数据连接请求，建立数据连接
  - 被动模式
    - 通过三次握手，建立控制连接；客户端的源端口是高位随机端口，目标端口是21端口；
    - 控制连接建立后，客户端进行身份验证，协商数据连接采用被动模式；随后客户端会向服务器发送PASV报 文，表示我们用被动模式
    - 服务端收到PASV报文，于是向客户端发送Port报文，表明自己监听的IP+端口
    - 客户端发起数据连接请求，建立数据连接

FTP协议有两种工作方式：PORT方式和PASV方式，中文意思为主动式和被动式。
在主动模式下，FTP客户端随机开启一个大于1024的端口N向服务器的21号端口发起连接，然后开放N+1号端口进行监听，并向服务器发出PORT N+1命令。服务器接收到命令后，会用其本地的FTP数据端口（通常是20）来连接客户端指定的端口N+1，进行数据传输。

在被动模式下，FTP库户端随机开启一个大于1024的端口N向服务器的21号端口发起连接，同时会开启N+1号端口。然后向服务器发送PASV命令，通知服务器自己处于被动模式。服务器收到命令后，会开放一个大于1024的端口P进行监听，然后用PORT P命令通知客户端，自己的数据端口是P。客户端收到命令后，会通过N+1号端口连接服务器的端口P，然后在两个端口之间进行数据传输。

- 主动模式被动模式的区别

总的来说，主动模式的FTP是指服务器主动连接客户端的数据端口，被动模式的FTP是指服务器被动地等待客户端连接自己的数据端口。
被动模式的FTP通常用在处于防火墙之后的FTP客户访问外界FTp服务器的情况，因为在这种情况下，防火墙通常配置为不允许外界访问防火墙之后主机，而只允许由防火墙之后的主机发起的连接请求通过。因此，在这种情况下不能使用主动模式的FTP传输，而被动模式的FTP可以良好的工作。



- 软件包
  - vsftpd
  - tftp（了解）

VSFTPD服务介绍

- 服务包：vsftpd
- 服务类型：由Systemd启动的守护进程
- 配置单元：`/usr/lib/systemd/system/vsftpd.service`
- 守护进程：`/usr/sbin/vsftpd`
- 端口：`21(ftp)`,`20(ftp-data)`
- 主配置文件：`/etc/vsftpd/vsftpd.conf`

- 用户访问控制配置文件：`/etc/vsftpd/ftpusers /etc/vsftpd/user_list`
- 日志文件：`/etc/logrotate.d/vsftpd`
- 配置文件参数

参数	作用
listen=NO	是否以独立运行的方式监听服务
listen_address=ip地址	设置要监听的IP地址
listen_port=21	设置FTP服务的监听端口
download_enable=YES	是否允许下载文件
userlist_enable=YES	设置用户列表为"允许"
userlist_deny=YES	设置用户列表为"禁止"
max_clients=0	最大客户端连接数，0为不限制
max_per_ip=0	同一IP地址的最大连接数，0为不限制
anonymous_enable=YES	是否允许匿名用户访问
anon_upload_enable=YES	是否允许匿名用户上传文件
anon_umask	匿名用户上传文件的umask
anon_root=/var/ftp	匿名用户的ftp根目录
anon_mkdir_write_enable=YES	是否允许匿名用户创建目录
anon_other_write_enable=YES	是否开放匿名用户的其他写入权限（重命名、删除等）
anon_max_rate=0	匿名用户的最大传输速率，0为不限制
local_enable=yes	是否允许本地用户登录
local_umask=022	本地用户上传文件的umask值
local_root=/vat/ftp	本地用户的ftp根目录
chroot_local_user=YES	是否将用户权限禁锢在ftp目录，以确保安全
local_max_rate=0	本地用户的最大传输速率，0为不限制

基础配置

- 安装vsftp

```
[root@localhost ~]#yum -y install vsftpd
```

- 准备分发的文件

```
[root@localhost ~]#touch /var/ftp/abc.txt
```

- 启动服务

```
[root@localhost ~]#systemctl start vsftpd
[root@localhost ~]#systemctl enable vsftpd
```

- 关闭防火墙

```
[root@localhost ~]#systemctl stop firewalld
[root@localhost ~]#setenforce 0
```

客户端工具

Linux中

- 第一种

```
[root@localhost ~]#yum install ftp -y
[root@localhost ~]#ftp <IP地址>
username:.......
password:.......
```

- 第二种

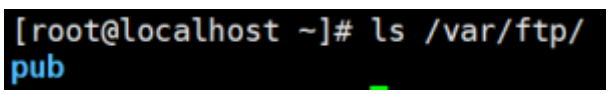
```
[root@localhost ~]#yum install lftp -y
[root@localhost ~]#lftp <IP地址>
```

- 区别
  - ftp工具是一定要输入用户名称和密码的，登录成功或者失败会给出提示。lftp不会直接给出登录成功或者失败的
  - 提示，需要输入ls工具才可以发现是否连接成功，优点在于连接更加方便

Windows中

- 第一种
  - 可以在浏览器、运行窗口或者资源管理器中输入`ftp://IP地址/`，这样访问的是ftp的根位置，如果需要访问相关
  - 目录可以输入`ftp://IP地址/目录/文件名`
- 第二种
  - 在DOS窗口中，输入命令`ftp <IP地址>`即可访问

> 需要注意的是直接访问ftp服务器的IP地址时访问的根位置目录是`/var/ftp`如下图，比如如果需要访问pub里的test可以访问`ftp://192.168.80.129/pub/test`。



案例1，匿名用户访问（默认开启）

```
[root@localhost ~]# vi /etc/vsftpd/vsftpd.conf
anonymous_enable=YES
anon_umask=022
anon_upload_enable=Yes
anon_mkdir_write_enable=Yes
```

```
anon_other_write_enable=Yes
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
[root@localhost ~]# systemctl restart vsftpd
```

案例2，本地用户访问

使用本地用户登录成功时位置在家目录的位置

```
[root@localhost ~]# vi /etc/vsftpd/vsftpd.conf
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
[root@localhost ~]# systemctl restart vsftpd
[root@localhost ~]# systemctl enable vsftpd
注意：出现在/etc/vsftpd/ftpuser  /etc/vsftpd/user_list这两个文件
中的内容将会被定义为黑名单
```

案例3，虚拟用户访问

1. 创建用于进行FTP认证的用户数据库文件，其中奇数行为账户名，偶数行为密码。

```
[root@localhost ~]# cd /etc/vsftpd/
[root@localhost vsftpd]# vi vuser.list
eagle
centos
cisco
centos
huawei
centos
```

2. 使用db_load命令用哈希（hash）算法将原始的明文信息文件转换成数据库文件
3. 降低数据库文件的权限（避免其他人看到数据库文件的内容）
4. 把原始的明文信息文件删除。

```
[root@localhost vsftpd]# db_load -T -t hash -f vuser.list vuser.db
[root@localhost vsftpd]# file vuser.db
vuser.db: Berkeley DB (Hash, version 9, native byte-order)
[root@localhost vsftpd]# chmod 600 vuser.db
[root@localhost vsftpd]# rm -f vuser.list
```

5. 创建一个本地用户，用来做虚拟用户在本地的代理，为了安全起见，禁止这个本 地用户登录

```
[root@localhost vsftpd]# useradd -d /var/ftproot -s /sbin/nologin virtual
[root@localhost vsftpd]# ls -ld /var/ftproot/
drwx------. 2 virtual virtual 59 8月  10 23:04 /var/ftproot/
[root@localhost vsftpd]# chmod -Rf 777 /var/ftproot/
```

6. 新建一个用于虚拟用户认证的PAM文件vsftpd.vu

```
[root@localhost vsftpd]# vi /etc/pam.d/vsftpd.vu
auth required pam_userdb.so db=/etc/vsftpd/vuser
account required pam_userdb.so db=/etc/vsftpd/vuser
```

7. 配置文件

```
[root@localhost vsftpd]# cat /etc/vsftpd/vsftpd.conf
anonymous_enable=NO
local_enable=YES
guest_enable=YES
guest_username=virtual
allow_writeable_chroot=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd.vu
userlist_enable=YES
tcp_wrappers=YES
[root@localhost vsftpd]#
```

8. 如果想要针对不同的用户设置不同的权限

```
[root@localhost vsftpd]# mkdir /etc/vsftpd/vusers_dir/
[root@localhost vsftpd]# cd /etc/vsftpd/vusers_dir/
[root@localhost vusers_dir]# touch huawei
[root@localhost vusers_dir]# vi eagle
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=YES
[root@localhost vusers_dir]#
[root@localhost vusers_dir]# vi /etc/vsftpd/vsftpd.conf
anonymous_enable=NO
local_enable=YES
guest_enable=YES
```

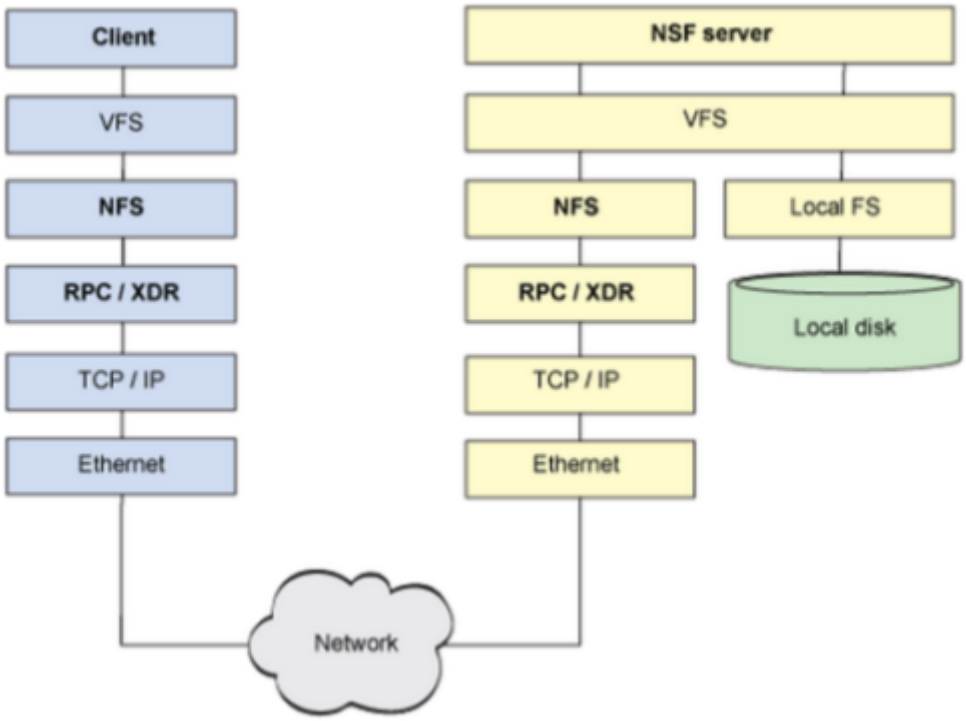
```
guest_username=virtual
allow_writeable_chroot=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd.vu
userlist_enable=YES
tcp_wrappers=YES
user_config_dir=/etc/vsftpd/vusers_dir
[root@localhost vusers_dir]# systemctl restart vsftpd
```

NFS(网络文件系统)

简介

- Linux/Unix系统之间共享文件系统的一种协议，通过网络让不同的主机之间共享文件或目录
- NFS的客户端主要为Linux
- 支持多节点同时挂载以及并发写入
- 提供文件共享服务
- 为集群中的web server配置后端存储

NFS协议模型



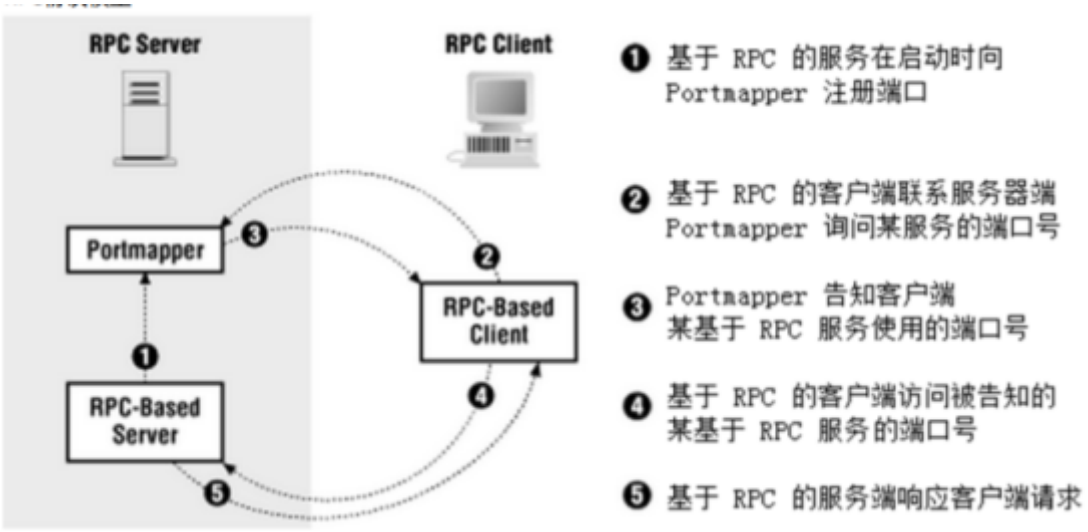
RPC远程过程调度

- NFS协议本身并没有网络传输功能，而是基于远程过程调用协议实现的
- 提供一个面向过程的远程服务的接口
- 可以通过网络从远程主机程序上请求服务，而不需要了解底层网络技术的协议
- 工作在OSI模型的会话层，它可以为遵从RPC协议应用层协议提供**端口注册**功能

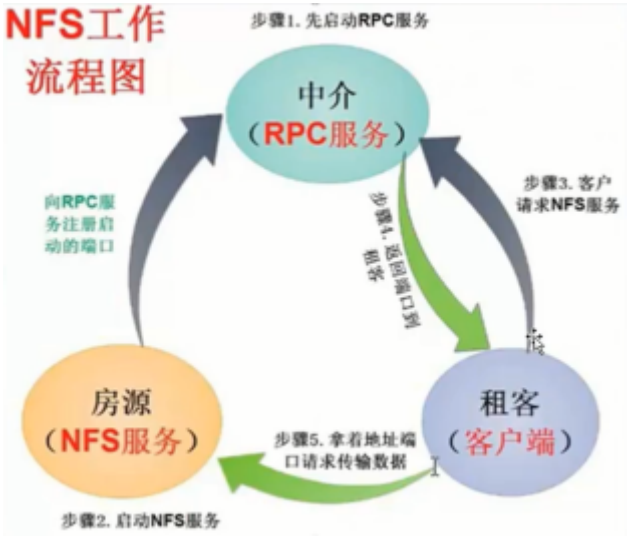
- 事实上，有很多服务（NFS和NIS等）都可以向RPC注册端口
- RPC使用网络端口111来监听客户端的请求

RPC协议模型

1. 基于rpc的服务（此处是指nfs服务，在别处有可能是代表其他服务）在启动时向portmapper注册端口
2. 基于rpc的客户端练习服务端portmapper询问服务的端口号
3. portmapper告知客户端某基于rpc服务使用的端口号
4. 基于rpc的客户端访问被告知单某基于rpc服务的端口
5. 基于rpc的服务响应客户端的请求

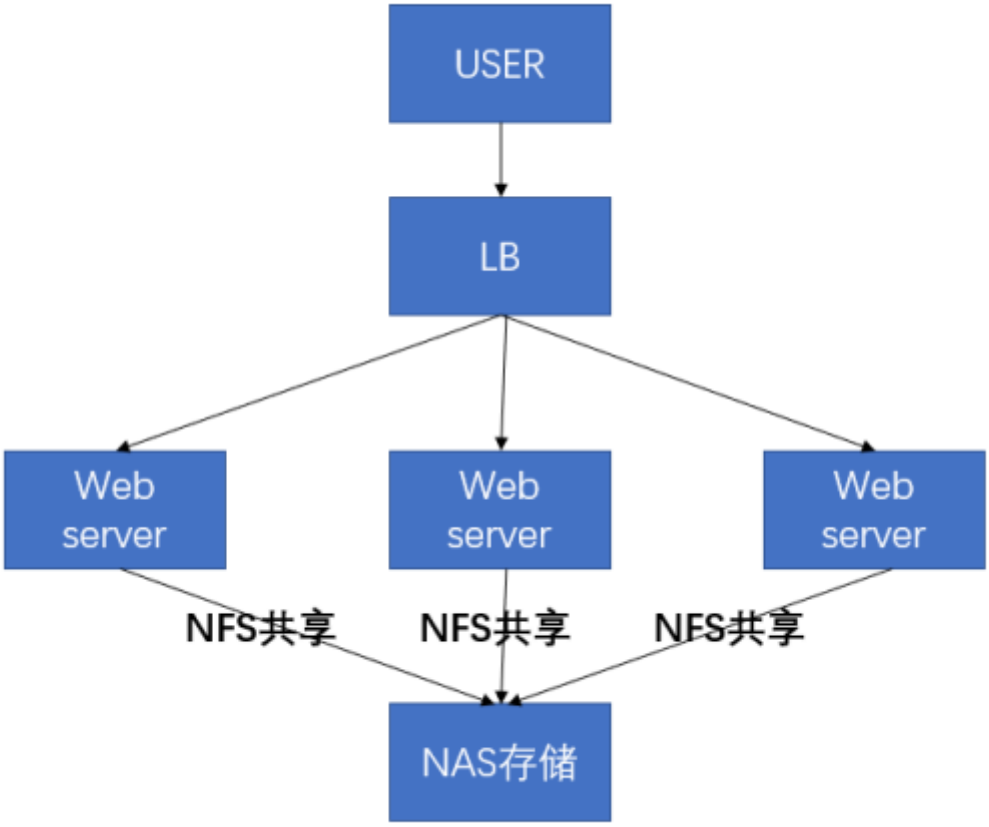


工作过程



案例，基于NFS搭建web后端NAS存储

用户访问负载均衡器节点（负载均衡的知识暂时不用关心），负载均衡器会将请求负载均衡得分发给web服务器，比如请求index.html界面，每一台服务器都会去NAS存储服务器上寻找想要被读取的数据。这可以大大降低服务器成本、运维成本（修改数据会牵一发而动全身）



- 关闭防火墙和selinux，每台机器操作一样

```
[root@server1 ~]# systemctl stop firewalld
[root@server1 ~]# setenforce 0
```

- 准备web

```
[root@server1 ~]# yum install httpd -y
[root@server1 ~]# systemctl start httpd

[root@server2 ~]# yum install httpd -y
[root@server2 ~]# systemctl start httpd
```

- 准备nas端

```
# 安装nfs服务
[root@server3 ~]# yum install -y nfs-utils
# 准备共享目录
[root@server3 ~]# mkdir /webdata
# 准备共享文件
[root@server3 ~]# echo "<h1>today i study but you sleep i good you bad</h1>" > /webdata/index.html
# 配置nfs服务
[root@server3 webdata]# cat /etc/exports
/webdata    192.168.80.0/24(rw)
[root@server3 ~]# systemctl start nfs-server.service
[root@server3 ~]# systemctl enable nfs-server.service
```

- 查看共享

```
[root@server1 ~]# yum install -y nfs-utils
[root@server1 ~]# showmount -e 192.168.80.153
[root@server2 ~]# yum install -y nfs-utils
[root@server2 ~]# showmount -e 192.168.80.153
```

- 挂载后端nas存储

```
[root@server1 ~]# mount -t nfs 192.168.80.153:/webdata /var/www/html/
[root@server1 ~]# df
文件系统                     1K-块     已用      可用 已用% 挂载点
/dev/mapper/centos-root 17811456 1153376 16658080    7% /
devtmpfs                   922468       0   922468    0% /dev
tmpfs                      933524       0   933524    0% /dev/shm
tmpfs                      933524    8804   924720    1% /run
tmpfs                      933524       0   933524    0% /sys/fs/cgroup
/dev/sda1                 1038336  145756   892580   15% /boot
tmpfs                      186708       0   186708    0% /run/user/0
192.168.80.153:/webdata 17811456 1082880 16728576    7% /var/www/html

[root@server2 ~]# mount -t nfs 192.168.80.153:/webdata /var/www/html/
[root@server2 ~]# df
文件系统                     1K-块     已用      可用 已用% 挂载点
/dev/mapper/centos-root 17811456 1127392 16684064    7% /
devtmpfs                   922468       0   922468    0% /dev
tmpfs                      933524       0   933524    0% /dev/shm
tmpfs                      933524    8836   924688    1% /run
tmpfs                      933524       0   933524    0% /sys/fs/cgroup
/dev/sda1                 1038336  145756   892580   15% /boot
tmpfs                      186708       0   186708    0% /run/user/0
192.168.80.153:/webdata 17811456 1082880 16728576    7% /var/www/html

```

- 最后在浏览器测试访问即可

```
nas存储端 133
web客户端 132
测试网站访问
关闭防火墙和selinux
配置nas
    安装nfs服务器
    yum install -y nfs-utils
    mkdir /webdata
    echo "nfs test..." > /webdata/index.html     准备测试页面
  配置nfs服务器
    vim /etc/exports
            /webdata    192.168..0/24(rw)
  启动nfs服务器
    systemctl start nfs-server
    systemctl enable nfs-server
    exportfs -v
检查是否输出正常
配置web客户端
    安装nfs客户端
    yum install -y nfs-utils httpd
    systemctl enable httpd
    systemctl start httpd
```

```
  查看存储的共享
    showmount -e 192.168..133                                    查询nfs服务器可用目录
  手动挂载
    mount -t nfs 192.168..133:/webdata /var/www/html
    umount /var/www/html                                         卸载挂载（需要的时候再使用）
  查看挂载
    df
```

NFS配置参数

参数	用途
rw	表示可读可写权限
ro	表示只读权限
Sync（同步传输数据）	请求或写入数据时，数据同步写入到NFS Server的硬盘后才返回。优点，数据安全不会丢；缺点，性能相对较差
Async（异步传输数据）	写入数据会先写到内存缓冲区中，直到硬盘有空挡才会再次写入磁盘，这样可以提升写入效率，但是如果出现服务器不正常关机，会导致缓冲区中的未写入磁盘的数据丢失。
no_root_squash	保持root用户不进行压缩。访问NFS Server共享目录的用户如果是root用户的话，它对该共享目录具有root权限，这个配置是为无盘客户端准备的，尽量少用。
root_squash	访问NFS Server共享目录的用户如果是root用户的话，它的权限会被压缩为匿名用户，会变成nfsnobody身份。
all_squash	不管访问NFS Server共享目录的用户身份如何，它的权限都将被压缩成匿名用户，同时它的 UID和GID都会变成nfsnobody 账号身份。在早期多个NFS客户端同时读写NFS Server数据时，这个参数很有用 在生产中配置NFS的重要技巧 1．确保所有客户端服务器对NFS 共享目录具备相同的用户访问权限。a. all_squash把所有客户端都压缩成固定的匿名用户（UID相同)b。就是anonuid, anongid指定的UID 和GID的用户。 2．所有的客户端和服务器端都需要有一个相同的UID和GID的用户，即nfsnobody (UID必须相同)
anonuid	指定的是匿名用户的uid(数字) 。 参数以 anon*开头即指anonymous匿名用户，这个用户的UID设置值通常为nfsnobody 的UID值，当然也可以自行设置这个UID值。但是，UID必须存在于/etc/passwd中。在多NFS Clients 时，如多台web server 共享一个 NFS目录，通过这个参数可以使得不同的NFS Clients写入的数据对所有NFS Clients保持同样的用户权限，即为配置的匿名UID对应用户权限，这个参数很有用，一般默认即可，
anongid	指定的是匿名用户组的gid(数字) 同anonuid，区别就是把uid《用户id)换成gid（组id)

自动挂载

可使用autofs服务按需要挂载外围设备，NFS共享等，并在空闲5分钟后后自动卸载

相关包和文件

- 软件包: autofs
- 服务文件: /usr/lib/systemd/system/autofs.service
- 配置文件: /etc/auto.master

配置文件格式

参看帮助: man 5 autofs

所有导出到网络中的NFS启用特殊匹配-host至"browse"

范例: /net目录可以自动挂载NFS共享

```
cat /etc/auto.master
/net    -hosts
cd /net/192.168.175.147
```

- 自动挂载资源有两种格式
  - 相对路径法:将mount point路径分成dirname和basename分别配置，可能会影响现有的目录结构
  - 绝对路径法:直接匹配全部绝对路径名称,不会影响本地目录结构

相对路径法

- `/etc/auto.master`格式

```
挂载点的dirname     指定目录的配置文件路径
```

- 指定目录的配置文件格式

```
挂载点的basename     挂载选项     选项设备
```

案例

- 相对路径法

```
vim /etc/auto.master
/misc    /etc/auto.misc
vim /etc/auto.misc
cd      -fstype=iso9660,ro,nosuid,nodev    :/dev/cdrom
```

- 相对路径法为支持通配符

```
vim /etc/auto.master
/misc    /etc/auto.misc
vim /etc/auto.misc
# 表示/misc下面的子目录和nfs共享/export目录的子目录同名
*      server:/export/&
```

绝对路径法

- /etc/auto.master格式

```
/-        指定配置文件路径
```

- 指定配置文件格式

```
绝对路径        挂载选项        选项设备
```

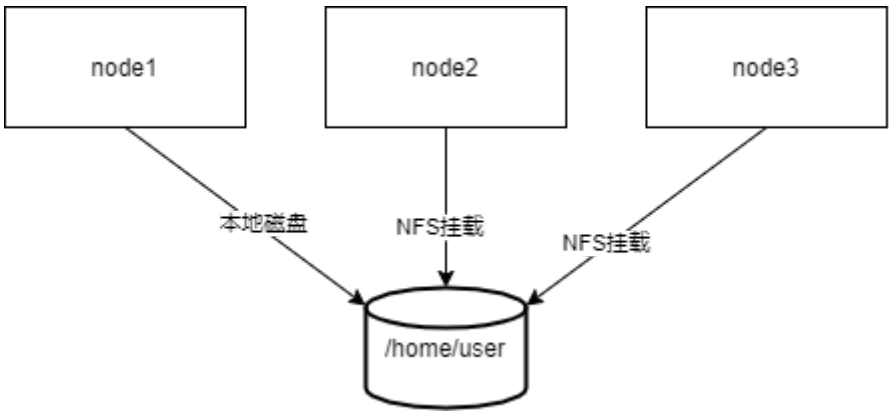
案例

- 绝对路径法

```
vim /etc/auto.master
/-      /etc/auto.direct
vim /etc/auto.direct
/nfsdir      -fstype=nfs    server1:/export/nfsdir
```

实战案例

将NFS的共享目录，通过autofs 发布出来，做为远程主机用户的家目录



环境准备

将node1中的用户家目录共享出来，node2在登陆这个用户的到时候，看到家目录下的文件是一致的

步骤

- NFS服务器创建用户和相应的目录，将用户user的家目录共享

```
[root@NFSserver ~]# mkdir /data
[root@NFSserver ~]# useradd -d /data/user user
[root@NFSserver ~]# id user
uid=1000(user) gid=1000(user) 组=1000(user)
[root@NFSserver ~]# vim /etc/exports.d/test.exports
/data/user    *(rw,anonuid=1000,anongid=1000,all_squash)
```

- 在nfs客户端上实现autofs

```
[root@NFSclient ~]# vim /etc/auto.master
/data   /etc/auto.home
[root@NFSclient ~]# vim /etc/auto.home
*    -fstype=nfs,vers=3    192.168.175.144:/data/user&
```

- 在nfs客户端上创建用户user

```
[root@NFSclient ~]# mkdir /data
[root@NFSclient ~]# useradd -d /data/user -u 1000 user
```

- 测试是否完成目标

```
[root@NFSserver ~]# su - user
[user@NFSserver ~]$ touch file
# 在NSF服务器上登录user用户，创建文件在家目录中
[root@NFSclient /]# su - user
[user@NFSclient ~]$ ll
总用量 0
-rw-rw-r--. 1 user user 0 4月  30 10:13 file
# 在NSF客户机上登录user用户，发现文件已经被共享了
```

SAMBA

简介

- 历史
  - 1987年，微软公司和英特尔公司共同制定了SMB（Server Messages Block，服务器消息块）协议，旨在解 决局域网内的文件或打印机等资源的共享问题，这也使得在多个主机之间共享文件变得越来越简单。到了1991年， 当时还在读大学的Tridgwell为了解决Linux系统与Windows系统之间的文件共享问题，基于SMB协议开发出了 SMBServer服务程序。这是一款开源的文件共享软件，经过简单配置就能够实现Linux系统与Windows系统之间的 文件共享工作。当时，Tridgwell想把这款软件的名字SMBServer注册成为商标，但却被商标局以SMB是没有意义的 字符而拒绝了申请。后来Tridgwell不断翻看词典，突然看到一个拉丁舞蹈的名字—Samba，而且这个热情洋溢的 舞蹈名字中又恰好包含了“SMB”，于是Samba服务程序的名字由此诞生。Samba服务程序现在已经成为在Linux系 统与Windows系统之间共享文件的最佳选择
- SMB协议
  - SMB（服务信息模块）协议是一个高层协议、它提供了在网络上的不同计算机之间共享文件，打印机和不同通 信资料的手段
  - SMB使用NetBIOS API实现面向连接的协议，该协议为Window是客户程序和服务提供了一个通过虚链路按照 请求——响应方式进行通信的机制
  - SMB的工作原理就是让NetBIOS与SMB协议运行在TCP/IP上，并且使用NetBIOS的名字解释器让Linux机器可 以在Windows的网络邻居中被看到，从而在Windows的网络邻居中被看到，从而和Windows9x/NT/200X进 行相互沟通，共享文件和打印机
- CIFS协议
  - 通用网际文件系统是微软服务器消息块协议（SMB）的增强版
  - 提供计算机用户在企业内部网和因特网上共享文件的标准方法
  - CIFS在TCP/IP运行，利用英特网上的全球域名服务系统（DNS）增强其可扩展性，同时为因特网上普遍存在 的慢速拨号连接优化

- 应用环境
  - 文件和打印机共享：文件和打印机共享是samba的主要功能，SMB进程实现资源共享，将文件和打印机发布到网络之中，以供用户可以访问
  - 身份验证和权限设置：samba服务支持user mode和domain mode 等身份验证和权限设置模式，通过加密方式可以保护共享的文件和打印机
  - 名称解析：samba通过NMB服务可以搭建NBNS服务器，提供域名解析，将计算机的NetBIOS名称解析为ip地址
  - 浏览服务，局域网中，samba可以成为本地主浏览服务器，保存可用资源列表，当使用客户端访问windows网上邻居时，会提供浏览列表，显示共享目录、打印机资源等
- 端口号：139和445
  - 在早期，SMB运行于NBT协议上，使用udp协议的137和138以及TCP协议的139端口

```
  cat /etc/services
    netbios-ns      137/tcp                         # NETBIOS Name Service
    netbios-ns      137/udp
    netbios-dgm     138/tcp                         # NETBIOS Datagram
Service
    netbios-dgm     138/udp
    netbios-ssn     139/tcp                         # NETBIOS session service
    netbios-ssn     139/udp
```

- NetBIOS协议
  - NetBIOS是Network Basic Input/Output System的简称，网络基本输入输出协议。协议，一般指能用于局域网通信的一套API，是由IBM公司开发的。主要作用，**通过NetBIOS协议获得计算机名称，然后把计算机名称解析为对应的IP地址**
- 基于C/S模式
- 安装

```
[root@server1 ~]# yum install samba -y
```

- 相关配置

```
/etc/sysconfig/samba：用于设置守护进程的启动参数。
/etc/samba/smb.conf：主配置文件。
/etc/samba/smbusers：用于映射Linux用户和Windows用户。
/etc/samba/lmhosts：用于设置NetBIOS名字与IP地址的对应关系表。
/etc/pam.d/samba：Samba的PAM配置文件
/etc/rc.d/init.d/smb：Samba的INIT启动脚本
```

- 相关工具

```
服务端工具：
/usr/bin/smbpasswd：用于设置Samba用户账号及口令
/usr/bin/testparm：用于检测配置文件的正确性
/usr/bin/smbstatus：用于查找网络中的Samba服务器
客户端工具：
/usr/bin/findsmb：用于查找网络中的Samba服务器
/usr/bin/smbclient：Linux下的Samba客户端
/usr/bin/smbget：基于SMB/CIFS的类似于wget的下载工具
/usr/bin/smbtar：类似于tar的归档工具，用于将SMB/CIFS的共享打包备份到Linux主机
```

- 主配置文件

```
[root@server1 ~]# cat /etc/samba/smb.conf
# 默认主配置文件：
[global] #全局参数。
    workgroup = MYGROUP #工作组名称
    server string = Samba Server Version %v #服务器介绍信息，参数%v为显示SMB版本号
    log file = /var/log/samba/log.%m #定义日志文件的存放位置与名称，参数%m为来访的主机名
    max log size = 50 #定义日志文件的最大容量为50KB
    security = user #安全验证的方式，总共有4种
    #share：来访主机无需验证口令；比较方便，但安全性很差
    #user：需验证来访主机提供的口令后才可以访问；提升了安全性
    #server：使用独立的远程主机验证来访主机提供的口令（集中管理账户）
    #domain：使用域控制器进行身份验证
    passdb backend = tdbsam #定义用户后台的类型，共有3种
    #smbpasswd：使用smbpasswd命令为系统用户设置Samba服务程序的密码
    #tdbsam：创建数据库文件并使用pdbedit命令建立Samba服务程序的用户
    #ldapsam：基于LDAP服务进行账户验证
    load printers = yes #设置在Samba服务启动时是否共享打印机设备
    cups options = raw #打印机的选项
[homes] #共享参数
    comment = Home Directories #描述信息
    browseable = no #指定共享信息是否在“网上邻居”中可见
    writable = yes #定义是否可以执行写入操作，与“read only”相反
[printers] #打印机共享参数
    comment = All Printers
    path = /var/spool/samba #共享文件的实际路径(重要)。
    browseable = no
    guest ok = no #是否所有人可见，等同于"public"参数。
    writable = no
    printable = yes
```

- 安全等级

```
Samba安全等级：
  User：由本地Samba服务器负责账户验证
  使用smbpasswd 设置账号（默认的安全等级）
  Domain：账户验证账户及口令的工作由其他的Windows 或Samba域控制器负责
  需要使用“password server”指令指定验证服务器
  Ads：验证账户及口令的工作由支持Kerberos验证的Windows活动目录服务器负责。
  需要使用“realm”指令指定Kerberos领域
  Share：匿名共享
Samba账户及口令数据库
1.Samba使用的账户文件/数据库是与系统账户文件分离的
2.当设置了user的安全等级后（默认），将由本地系统对问Samba共享资源的用户进行认证
3.用.tdb格式的口令数据库，初始情况下口令数据库文件并不存在
4.为了创建Samba的口令数据库文件，管理员可以在添加Samba账户的同时创建它
5.管理员可以使用smbpasswd命令配置Samba账号并设置其口令
```

配置一个共享资源(具体步骤)

1. 设置共享名称
    1. 共享资源发布后，必须为每个共享目录或打印机设置不同的共享名称，给网络用户访问时使用，并且共享名可以与原目录名不同。例如，samba服务器上有个目录为/share，需要发布该目录为共享目录，定义共享名为public

2. 共享资源描述
    1. 格式：`comment = 备注信息`
    2. 备注信息通常是用来进行解释说明的
3. 共享路径
    1. 共享资源的原始完整路径
    2. 格式：`path = 绝对路径`
4. 设置匿名访问
    1. 共享资源如果对匿名访问进行设置，可以更改public字段
    2. 格式：`public = yes | no （yes代表允许匿名访问，no代表不允许）`
5. 设置访问用户
    1. 如果共享资源存在重要数据的话，需要访问用户审核，可以使用valid users字段进行设置
    2. 格式：
        1. `valid users = 用户`
        2. `valid users = @组名`
6. 设置目录只读
    1. 共享目录如果限制用户的读写操作，可以通过readonly实现
    2. 格式：`readonly = yes | no`
        1. yes代表只读
        2. no代表读写
7. 设置目录可写
    1. 如果目录允许用户写操作，可以使用writable或write list两个字段进行设置
    2. 格式：
        1. `writable = yes` 读写
        2. `writable = no` 只读
        3. `write list = 用户名`
        4. `write list = @ 组名`

案例，通过用户名共享文件

共享销售部`/xsb`这个目录，只有知道用户名和密码的同时可以看这个共享，在/xsb目录中存放销售部重要的数据。需要将security设置为user级别，这样可以启用samba身份验证机制，然后在共享目录`/xsb`下设置valid user 字段，配置只允许销售部员工能访问这个共享目录

- 修改主配置文件安全相关设置

```
[root@server1 ~]# vim /etc/samba/smb.conf
[global]
        workgroup = SAMBA
        security = user

#       passdb backend = tdbsam
        passdb backend = smbpasswd
        smb passwd file = /etc/samba/smbpasswd
        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw
# 重启smb服务之后，会自动生成/etc/samba/smbpasswd该文件

```

- 添加销售部用户和组

```
[root@server1 ~]# groupadd xsb
[root@server1 ~]# useradd -g xsb -M -s /sbin/nologin xsb01
[root@server1 ~]# useradd -g xsb -M -s /sbin/nologin xsb02
[root@server1 ~]# useradd jsb01
```

- 添加相应的samba账户

```
[root@server1 ~]# smbpasswd -a xsb01
New SMB password:
Retype new SMB password:
Added user xsb01.
[root@server1 ~]# smbpasswd -a xsb02
New SMB password:
Retype new SMB password:
Added user xsb02.
```

- 指定共享目录

```
[root@server1 ~]# mkdir /xsb
[root@server1 ~]# cp /etc/hosts /xsb
[root@server1 ~]# vim /etc/samba/smb.conf
[xsb]
        comment = Xsb Data
        path = /xsb
        valid user = xsb01,xsb02
```

- 重启服务

```
[root@server1 ~]# systemctl restart smb.service
[root@server1 ~]# systemctl restart nmb.service
```

- 检查139和445端口号

```
[root@server1 ~]# ss -tanl
State       Recv-Q Send-Q  Local Address:Port                 Peer
Address:Port
LISTEN      0      50                  *:139                             *:*

LISTEN      0      50                  *:445                             *:*

LISTEN      0      50                 :::139                            :::*

LISTEN      0      50                 :::445                            :::*
```

- 客户端验证

```
# linux上验证
[root@server2 ~]# yum install samba-client -y
[root@server2 ~]# smbclient -L //192.168.80.151/xsb -U xsb01
Enter SAMBA\xsb02's password:

```

```
    Sharename       Type      Comment
    ---------       ----      -------
    print$          Disk      Printer Drivers
    xsb             Disk      Xsb Data
    IPC$            IPC       IPC Service (Samba 4.10.16)
    xsb02           Disk      Home Directories
Reconnecting with SMB1 for workgroup listing.
    Server               Comment
    ---------            -------

    Workgroup            Master
    ---------            -------
    SAMBA                SERVER1
# 在windows上进行验证
    windows验证：
    在Window运行输入地址：\\192.168.10.10
    用户名：******
    密码：******
    可以在DOS窗口中使用命令net use * /delete 清空用户缓存信息
```

- 在Linux上进行挂载

```
[root@server2 ~]# mkdir /xsbdata
[root@server2 ~]# yum install cifs-utils -y
[root@server2 ~]# vim auth.smb
username=xsb01
password=1
[root@server2 ~]# vim /etc/fstab
//192.168.80.151/xsb /xsbdata   cifs    defaults,credentials=/root/auth.smb
    0 0
[root@server2 ~]# df
文件系统                    1K-块     已用      可用 已用% 挂载点
/dev/mapper/centos-root 17811456 1099604 16711852    7% /
devtmpfs                  922468       0   922468    0% /dev
tmpfs                     933524       0   933524    0% /dev/shm
tmpfs                     933524    8852   924672    1% /run
tmpfs                     933524       0   933524    0% /sys/fs/cgroup
/dev/sda1                1038336  145756   892580   15% /boot
tmpfs                     186708       0   186708    0% /run/user/0
//192.168.80.151/xsb    17811456 1108900 16702556    7% /xsbdata
[root@server2 ~]# ls /xsbdata/
hosts
```

```
扩展
隐藏目录
  隐藏共享目录，访问时必须输入绝对的URL进行访问
  格式：browseable = no
控制访问源
    使用hosts allow 和 hosts deny进行控制访问源
  hosts allow = 192.168.1.                          允许192.168.1.0/24网段进行访问
  hosts deny = 192.168.1.3                          禁止192.168.1.3主机进行访问
  当allow和deny同时设置时allow优先，但上述情况无法拒绝192.168.1.3，这种情况可以使用
except字段
```

```
  hosts allow = 192.168.1. EXCEPT   192.168.1.3
写入控制
    writable = yes
  write list =xsb01                                    开启samba服务的写入权限
  chmod 777 /xsb                                       给共享目录赋予写入
权限
用户账号映射（为了防止系统账号被黑客破解）
  username map = /etc/samba/smbusers  全局添加username map字段
  useradd rm
  passwd rm
  vim /etc/samba/smbusers
    rm = testuser
   重启smb服务
```