**本节所讲内容：**

1. **VSFTP服务器概述**
2. **安装配置VSFTP**
3. **实战：匿名访问VSFTP**
4. **实战：用户名密码方式访问VSFTP**
5. **NFS概述-配置NFS服务器并实现开机自动挂载**

**FTP服务端：xuegod63.cn IP：192.168.0.63**

**FTP客户端：xuegod64.cn IP：192.168.0.64**

**FTP服务概述：**

**FTP服务器（File Transfer Protocol Server）是在互联网上提供文件**[**存储**](http://baike.baidu.com/view/87682.htm)**和访问服务的计算机，它们依照FTP协议提供服务。**

**FTP（File Transfer Protocol: 文件传输协议）作用： Internet 上用来传送文件的协议**

**常见FTP服务器：**

**windows：Serv-U FTP Server，filezilla\_server**

**Linux：ProFTPD:（Professional FTP daemon）一个Unix平台上或是类Unix平台上（如Linux, FreeBSD等）的FTP服务器程序。**

**今天的主角：vsftp**

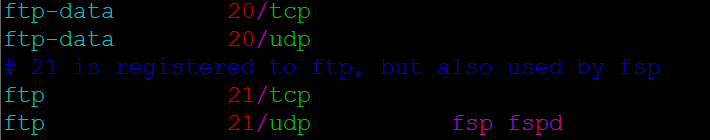
**VSFTP是一个基于GPL发布的类Unix系统上使用的**[**FTP服务器**](http://baike.baidu.com/view/106754.htm)**软件，它的全称是Very Secure FTP 从此名称可以看出来，编制者的初衷是代码的安全。**

**特点：**

**它是一个安全、高速、稳定的FTP服务器；**

**模式： C/S 模式**

**端口：**

**[root@xuegod64 ~]# vim /etc/services **

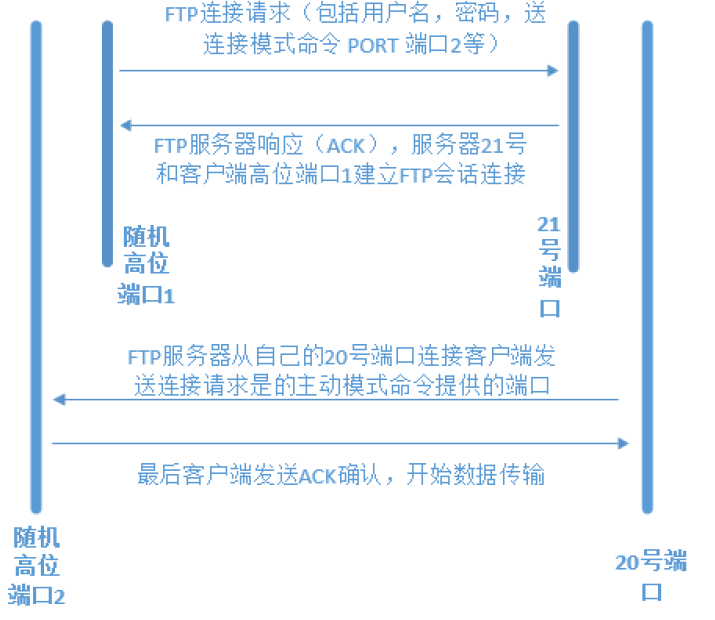
**20 (传数据) 21 (传指令)**

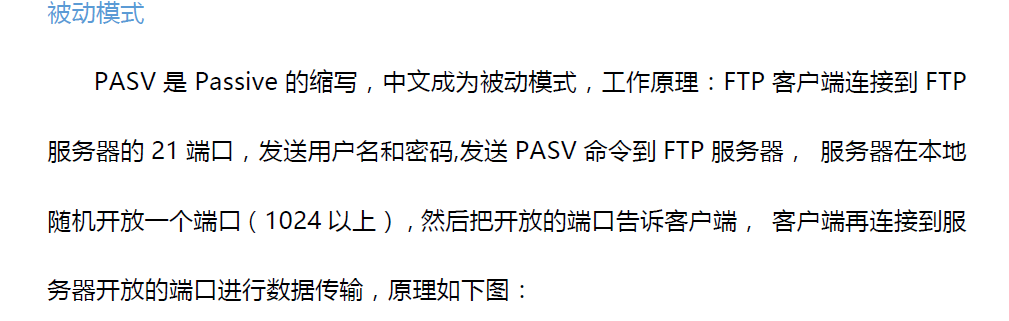
**fsp（File Service Protocol）： 文件服务协议。**

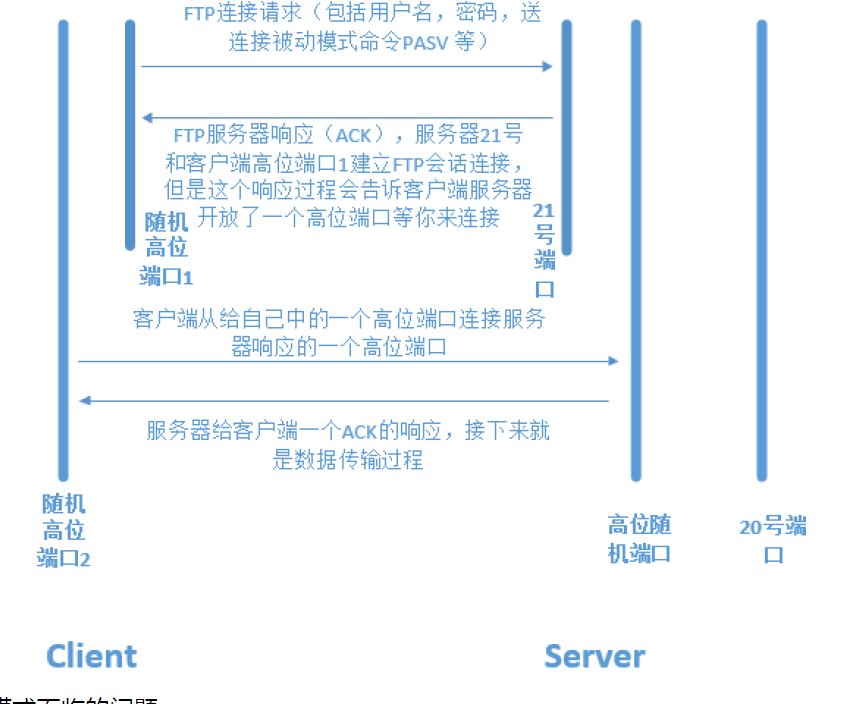
**工作流程（原理）：**

**##面试经常会遇到这样问题，大家需要注意下**

****







**这里的主动和被动，是相对于的FTP server 端来判断的**

**如果server 去连接client 开放的端口，说明是主动的，相反，如果client去连接server开放的端口，则是被动的。**

**安装vsftp服务器端、客户端**

**[root@xuegod63 ~]# rpm -ivh /mnt/Packages/vsftpd-2.2.2-6.el6\_0.1.x86\_64.rpm**

**或**

**yum -y install vsftpd lftp**

**安装客户端：**

**[root@xuegod64 ~]# yum install -y lftp**

**lftp**

**注：从RHEL6开始，系统镜像中默认没有ftp客户端命令。取而代之的是lftp命令**

**Linux客户端：**

**lftp 是一个功能强大的下载工具，它支持访问文件的协议: ftp, ftps, http, https, hftp, fish.(其中ftps和https需要在编译的时候包含openssl库)。llftp的界面非常好一个shell: 有命令补全，历史记录，允许多个后台任务执行等功能，使用起来非常方便。它还有书签、排队、镜像、断点续传、多进程下载等功能。**

**配置文件位置：**

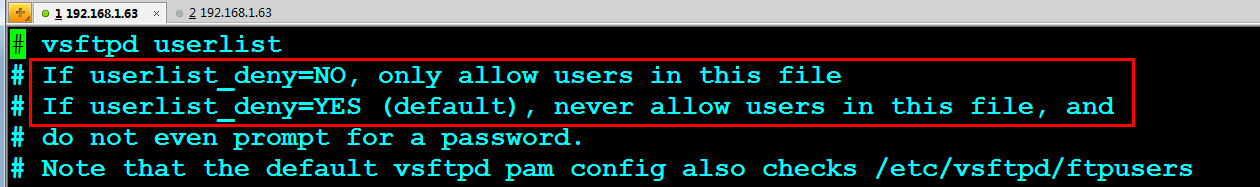
**2.vsftpd 相关文档**

**/etc/vsftpd/vsftpd.conf：vsftpd 的核心配置文件**

**/etc/vsftpd/ftpusers：用于指定哪些用户不能访问FTP 服务器。 黑名单**

**/etc/vsftpd/user\_list：指定允许使用vsftpd 的用户列表文件。 白名单**

**vim /etc/vsftpd/user\_list**



**# 如果userlist\_deny= YES（默认），绝不允许在这个文件中的用户登录ftp，甚至不提示输入密码**

**#prompt 提示**

**/etc/vsftpd/vsftpd\_conf\_migrate.sh：是vsftpd 操作的一些变量和设置脚本**

**/var/ftp/：默认情况下匿名用户的根目录**

**启动服务**

**[root@xuegod63 vsftpd]# systemctl start vsftpd**

**[root@xuegod63 vsftpd]# systemctl enable vsftpd**

**##注意关闭iptables 和selinux**

**[root@xuegod63 ~]# netstat -antup | grep ftp**

**tcp 0 0 0.0.0.0:21 0.0.0.0:\* LISTEN 11725/vsftpd**

**互动： 为什么看不到20端口？。**

**因为没有数据通信。**

**此服务的使用方法**

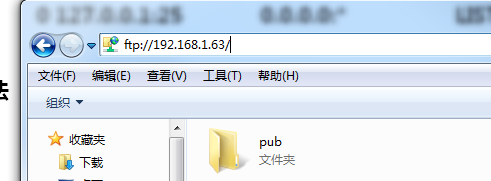
**Linux：**

**[root@xuegod63 vuserconfig]# rpm -vih /mnt/Packages/lftp-4.0.9-1.el6.x86\_64.rpm**

**[root@xuegod63 vsftpd]# lftp 192.168.0.63**

**windows：**

**通过浏览器访问或 打开文件夹，在地址栏输入地址：ftp://192.168.0.63/**



**修改配置文件，实战举例**

**例1：公司技术部准备搭建一台功能简单的FTP 服务器，允许所有员工上传和下载文件，并允许创建用**

**户自己的目录。**

**分析：**

**允许所有员工上传和下载文件需要设置成允许匿名用户登录并且需要将允许匿名用户上传功能**

**开启，**

**anon\_mkdir\_write\_enable 字段可以控制是否允许匿名用户创建目录。**

**[root@xuegod63 vsftpd]# cp vsftpd.conf vsftpd.conf.back**

**vim /etc/vsftpd/vsftpd.conf #修改以下内容**

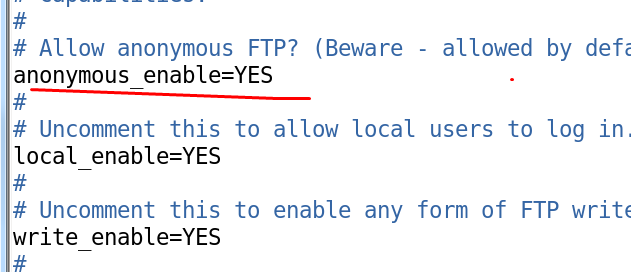
**允许匿名用户访问**

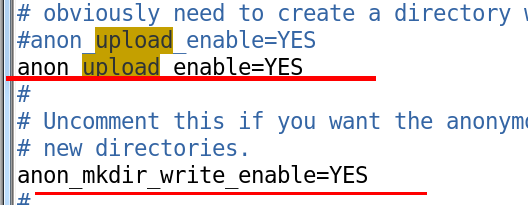
**anonymous\_enable=YES**

**允许匿名用户上传文件并可以创建目录**

**anon\_upload\_enable=YES**

**anon\_mkdir\_write\_enable=YES**





**启动服务：**

**service vsftpd restart**

**测试： 不能写**



**[root@xuegod63 vsftpd]# chown** [**ftp.ftp**](ftp://ftp.ftp) **/var/ftp/pub/**

**然后测试，是可以新建文件夹了，但是不能重命名，不能删除！**

**[root@xuegod63 vsftpd]# vim vsftpd.conf**

**anon\_other\_write\_enable=YES ##默认没有，需要手动添加下这行**

**重启服务，即可重命名文件夹。**

**但是可以删除文件夹了，这个参数对匿名用户来说权限太大,不安全，均衡使用这个参数**

**注意，默认匿名用户家目录的权限是755，这个权限是不能改变的。切记！**

**下面我们来一步一步的实现,先修改目录权限,创建一个公司上传用的目录，叫xuegoddata,设置拥有者为ftp 用户所有，目录权限是755**

**[root@xuegod63 vsftpd]# mkdir /var/ftp/xuegoddata**

**[root@xuegod63 vsftpd]# chown ftp.ftp /var/ftp/xuegoddata/**

**[root@xuegod63 vsftpd]# ll -d !$**

**ll -d /var/ftp/xuegoddata/**

**drwxr-xr-x 2 ftp root 4096 Mar 9 19:30 /var/ftp/xuegoddata/**

**然后重新启动服务**

**[root@xuegod63 ~]# service vsftpd restart**

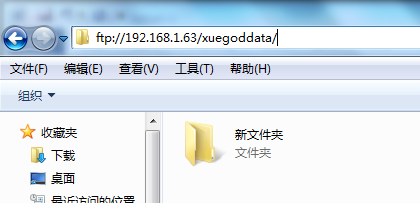
**Shutting down vsftpd: [ OK ]**

**Starting vsftpd for vsftpd: [ OK ]**

**测试**

**匿名登录FTP**

**现在我们匿名上传**



**现在匿名上传的文件是禁止删除滴~**



**这样匿名用户的上传就算成功了**

**注：工作中，匿名用户只是只读访问，写的权限也没有的。**

**例2：**

**公司内部现在有一台FTP 和WEB 服务器，FTP 的功能主要用于维护公司的网站内容，包括上传文**

**件、创建目录、更新网页等等。公司现有两个部门负责维护任务，他们分别适用team1 和team2**

**帐号进行管理。先要求仅允许team1 和team2 帐号登录FTP 服务器，但不能登录本地系统，并将**

**这两个帐号的根目录限制为/var/www/html，不能进入该目录以外的任何目录。**

**ftp 和www web服务器相结合。**

**www web服务器根目录： /var/www/html**

**只允许：team1和team2两用户 可以上传。 vsftp禁止匿名。**

**分析：**

**将FTP 和WEB 服务器做在一起是企业经常采用的方法，这样方便实现对网站的维护，为了增强安**

**全性，首先需要使用仅允许本地用户访问，并禁止匿名用户登录。其次使用chroot 功能将team1**

**和team2 锁定在/var/www/html 目录下。如果需要删除文件则还需要注意本地权限**

**解决方案：**

**（1）建立维护网站内容的ftp 帐号team1 和team2 并禁止本地登录，然后设置其密码**

**[root@xuegod63 ~]# useradd -s /sbin/nologin team1**

**[root@xuegod63 ~]# useradd -s /sbin/nologin team2**

**[root@xuegod63 ~]# echo "123456" | passwd --stdin team1**

**Changing password for user team1.**

**passwd: all authentication tokens updated successfully.**

**[root@xuegod63 ~]# echo "123456" | passwd --stdin team2**

**Changing password for user team2.**

**passwd: all authentication tokens updated successfully.**

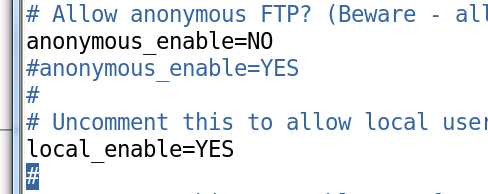
**2）配置vsftpd.conf 主配置文件并作相应修改**

**[root@xuegod63 vsftpd]# cp vsftpd.conf.back vsftpd.conf**

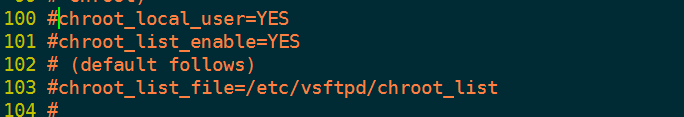
**vim /etc/vsftpd/vsftpd.conf**

**anonymous\_enable=NO：禁止匿名用户登录**

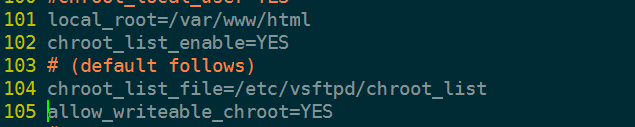
**local\_enable=YES：允许本地用户登录**



**改：**

****

**为：**

****

**local\_root=/var/www/html：设置本地用户的根目录为/var/www/html**

**chroot\_list\_enable=YES：激chroot 功能**

**chroot\_list\_file=/etc/vsftpd/chroot\_list：设置锁定用户在根目录中的列表文件。此文件存放要锁定的用户名**

**allow\_writeable\_chroot=YES :允许锁定的用户有写的权限**

**保存退出**

**（3）建立/etc/vsftpd/chroot\_list 文件，添加team1 和team2 帐号**

**[root@xuegod63 vsftpd]# touch /etc/vsftpd/chroot\_list**

**[root@xuegod63 ~]# ll !$**

**ll /etc/vsftpd/chroot\_list**

**-rw-r--r-- 1 root root 0 Nov 10 17:08 /etc/vsftpd/chroot\_list**

**[root@xuegod63 ~]# vim /etc/vsftpd/chroot\_list #写入以下内容，一行，一个用户名**

**team1**

**team2**

**（5）修改本地权限**

**[root@xuegod63 ~]# ll -d /var/www/html/**

**drwxr-xr-x. 2 root root 4096 Oct 6 2011 /var/www/html/**

**[root@xuegod63 ~]# chmod -R o+w /var/www/html/**

**[root@xuegod63 ~]# ll -d /var/www/html/**

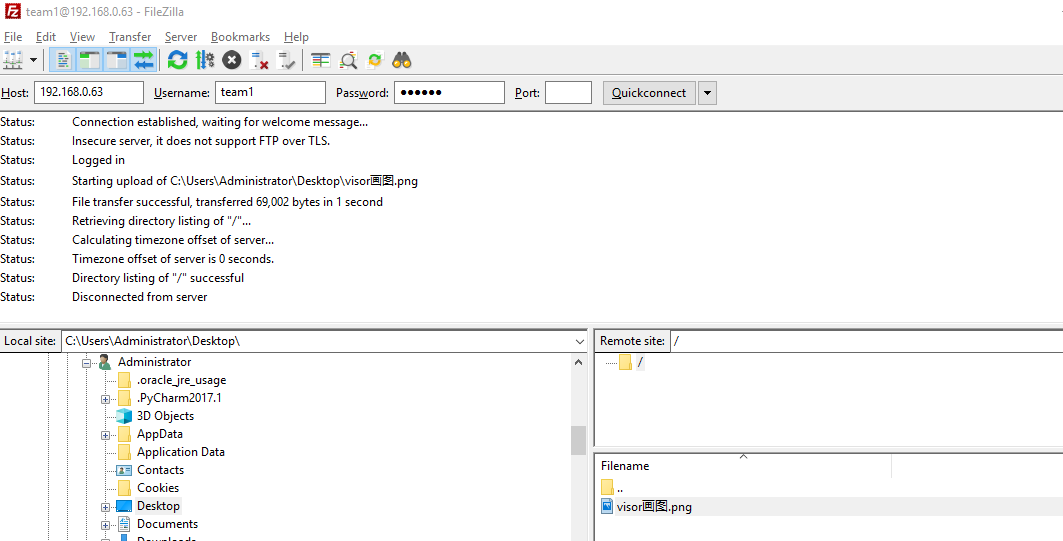
**drwxr-xrwx. 2 root root 4096 Oct 6 2011 /var/www/html/**

**（6）重启vsftpd 服务使配置生效**

**service vsftpd restart**

**（7）测试**

**测试：**

****

**客户端用lftp登陆查看：**

**root@xuegod64 ~]# lftp 192.168.0.63 -u team1,123456**

**lftp team1@192.168.0.63:~> ls**

**-rw-r--r-- 1 0 0 1384 Jul 30 01:56 passwd**

**lftp team1@192.168.0.63:/>**

**补充: 配置vsftpd,使用SSL证书加密数据传输**

**FTP与HTTP一样缺省状态都是基于明文传输，希望FTP服务器端与客户端传输保证安全，可以为FTP配置SSL**

**1, 使用OpenSSL生成自签证书**

**[root@xuegod63 vsftpd]# openssl req -new -x509 -nodes -out vsftpd.pem -keyout vsftpd.pem -days 3560**

**Generating a 2048 bit RSA private key**

**..........................................................................................+++**

**.....................+++**

**writing new private key to 'vsftpd.pem'**

**-----**

**You are about to be asked to enter information that will be incorporated**

**into your certificate request.**

**What you are about to enter is what is called a Distinguished Name or a DN.**

**There are quite a few fields but you can leave some blank**

**For some fields there will be a default value,**

**If you enter '.', the field will be left blank.**

**-----**

**Country Name (2 letter code) [XX]:ZH**

**State or Province Name (full name) []:JS**

**Locality Name (eg, city) [Default City]:NJ**

**Organization Name (eg, company) [Default Company Ltd]:XS**

**Organizational Unit Name (eg, section) []:XS**

**Common Name (eg, your name or your server's hostname) []:XS.COM**

**Email Address []:XS@QQ.COM**

**OpenSSL 简单参数解释:**

**req - 是 X.509 Certificate Signing Request （CSR，证书签名请求）管理的一个命令。**

**x509 - X.509 证书数据管理。**

**days - 定义证书的有效日期。**

**newkey - 指定证书密钥处理器。**

**keyout - 设置密钥存储文件。**

**out - 设置证书存储文件，注意证书和密钥都保存在一个相同的文件**

**2,创建证书文件存放目录**

**[root@xuegod63 vsftpd]# mkdir .sslkey**

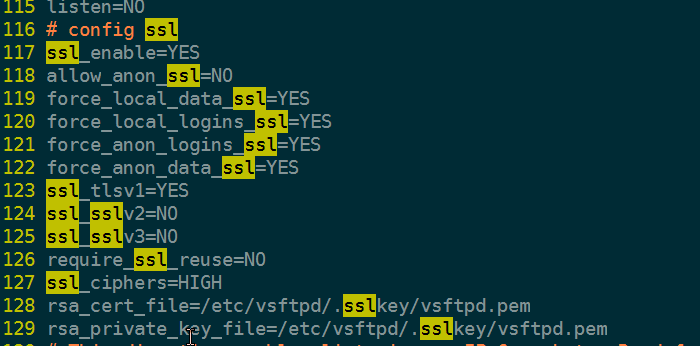
**[root@xuegod63 vsftpd]# cp vsftpd.pem .sslkey/**

**[root@xuegod63 vsftpd]# chmod 400 .sslkey/vsftpd.pem**

**3, 修改配置文件,支持SSL**

**[root@xuegod63 vsftpd]# vim vsftpd.conf**

**添加如下配置:**

****

**ssl\_enable=YES #启用SSL支持**

**allow\_anon\_ssl=NO**

**force\_local\_data\_ssl=YES**

**force\_local\_logins\_ssl=YES**

**force\_anon\_logins\_ssl=YES**

**force\_anon\_data\_ssl=YES**

**#上面四行force 表示强制匿名用户使用加密登陆和数据传输**

**ssl\_tlsv1=YES #指定vsftpd支持TLS v1[**

**ssl\_sslv2=YES #指定vsftpd支持SSL v2**

**ssl\_sslv3=YES #指定vsftpd支持SSL v3**

**require\_ssl\_reuse=NO #不重用SSL会话,安全配置项**

**ssl\_ciphers=HIGH #允许用于加密 SSL 连接的 SSL 算法。这可以极大地限制那些尝试发现使用存在缺陷的特定算法的攻击者**

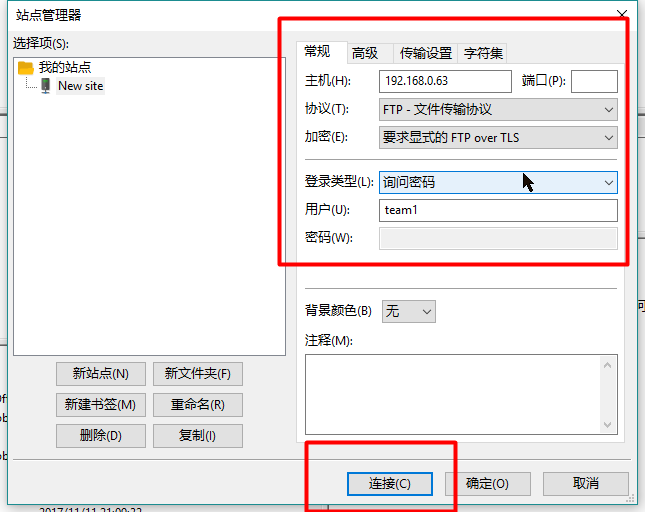
**rsa\_cert\_file=/etc/vsftpd/.sslkey/vsftpd.pem**

**rsa\_private\_key\_file=/etc/vsftpd/.sslkey/vsftpd.pem**

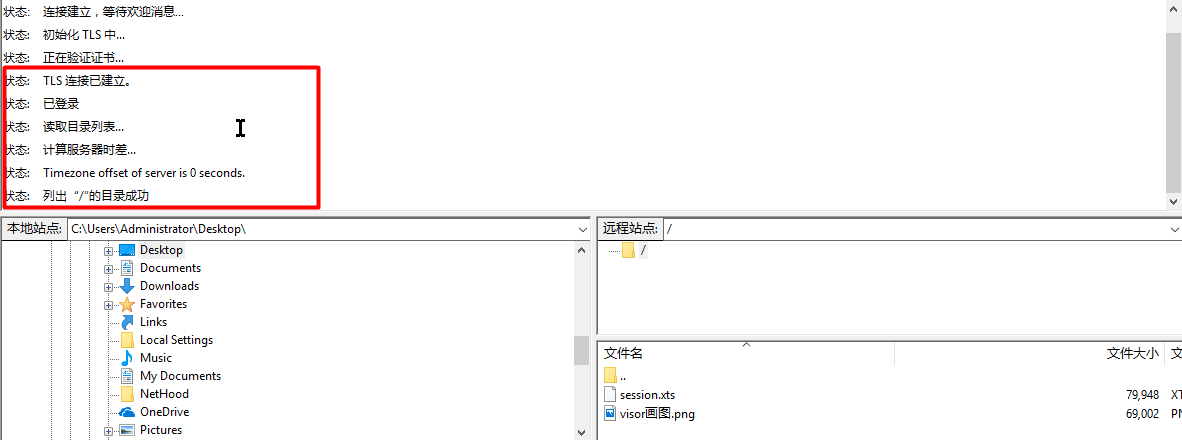
**#定义 SSL 证书和密钥文件的位置**

**注意:上面的配置项不要添加到vsftpd.conf 文件最后,否则启动报错**

**4,配置FileZilla客户端验证:**

****

****

****

**连接成功,发现可以使用TLS加密传输了**

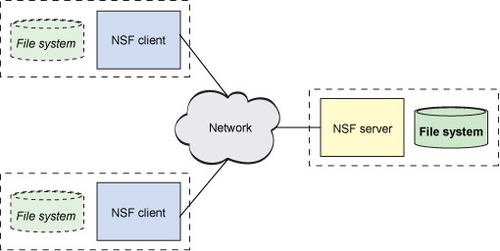
**注意: 在工作中,内网FTP传输,可以不用证书加密传输**

**如果FTP服务器在公网,为了数据的安全性,就一定要配置证书加密传输**

**NFS概述-配置NFS服务器并实现开机自动挂载**

**NFS服务端概述：**

**NFS，是Network File System的简写，即**[**网络文件系统**](http://baike.baidu.com/view/544339.htm)**。网络文件系统是FreeBSD支持的文件系统中的一种，也被称为NFS. NFS允许一个系统在网络上与他人共享目录和文件。通过使用NFS，用户和程序可以像访问本地文件一样访问远端系统上的文件。**



**模式： C/S 模式**

**端口：**

**RHEL7是以NFSv4作为默认版本，NFSv4使用TCP协议（端口号是2049）和NFS服务器建立连接**



**安装nfs**

**[root@xuegod63 ~]# yum -y install rpcbind nfs-utils**

**配置文件位置**

**[root@xuegod63 ~]# ls /etc/exports**

**/etc/exports**

**启动NFS服务**

**先查看2049端口是否开放：**

**[root@xuegod63 ~]# netstat -antpu | grep 2049**

**[root@xuegod63 ~]# systemctl start rpcbind**

**[root@xuegod63 ~]# systemctl start nfs-server.service**

**再次查看端口监听状态**

**[root@xuegod63 ~]# netstat -antpu | grep 2049**

**tcp 0 0 0.0.0.0:2049 0.0.0.0:\* LISTEN -**

**tcp 0 0 :::2049 :::\* LISTEN +**

**配置开机自动启动**

**[root@xuegod63 ~]# chkconfig nfs-server on**

**服务的使用方法**

**showmount -e NFS服务器IP**

**例：**

**[root@xuegod64 ~]# showmount -e 192.168.0.63**

**Export list for 192.168.0.63:**

**挂载**

**[root@xuegod64 ~]# mount 192.168.0.63:/tmp /opt**

**修改配置文件，实战举例**

**[root@xuegod63 ~]# vim /etc/exports**

**/media \*(rw)**

**注意: \* 表示对所有网段开放权限**

**也可以指定特定的网段**

**重启服务**

**[root@xuegod63 ~]# exportfs -rv ##重新读取配置文件,不中断服务.**

**客户端查看：**

**[root@xuegod64 ~]# showmount -e 192.168.0.63**

**Export list for 192.168.0.63:**

**/media \***

**挂载共享**

**[root@xuegod64 ~]# mount -t nfs 192.168.0.63:/media/ /opt/**

**[root@xuegod64 ~]# df -h**

**Filesystem Size Used Avail Use% Mounted on**

**/dev/sda2 9.7G 4.0G 5.2G 44% /**

**tmpfs 996M 80K 996M 1% /dev/shm**

**/dev/sda1 485M 39M 421M 9% /boot**

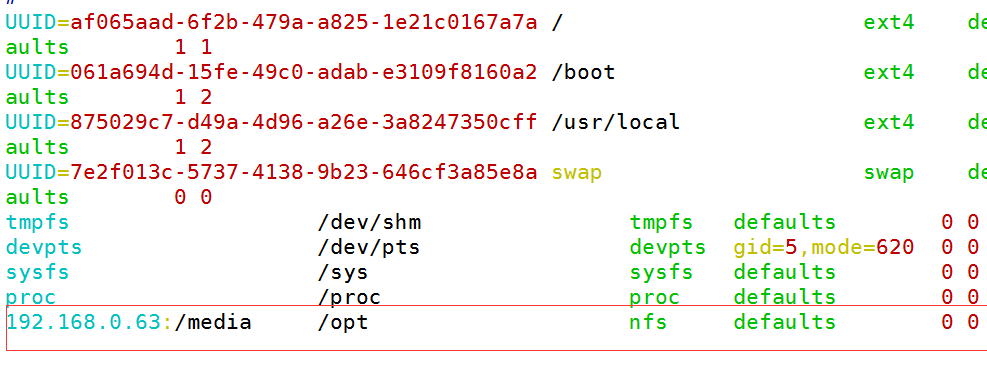
**/dev/sr0 3.7G 3.7G 0 100% /mnt**

**192.168.0.63:/media/ 9.7G 4.0G 5.3G 43% /opt**

**开机自动挂载：**

**编辑 [root@xuegod63 ~]# vim /etc/fstab**

**在文件最后添加自动挂载的信息:**



**验证写入权限**

**[root@xuegod64 ~]# touch /opt/a.txt**

**touch: 无法创建"/opt/a.txt": 权限不够**

**解决方法：**

**设置访问权限一般包含2部分**

**1）服务本身权限**

**2）目录访问权限**

**nfs默认使用nfsnobody用户**

**[root@xuegod63 ~]# grep nfs /etc/passwd**

**rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin**

**nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin**

**修改权限**

**[root@xuegod63 ~]# chmod 777 -R /media/**

**或**

**[root@xuegod63 ~]# chown nfsnobody.nfsnobody -R /media/**

**再次验证写入权限**

**[root@xuegod64 ~]# touch /opt/a.txt**

**[root@xuegod64 ~]# ll !$**

**ll /opt/a.txt**

**-rw-r--r-- 1 nfsnobody nfsnobody 0 5月 24 2016 /opt/a.txt**

**下面是一些NFS共享的常用参数：**

**ro                    只读访问   
 rw                   读写访问   
 sync               资料同步写入到内存与硬盘当中  
 async             资料会先暂存于内存当中，而非直接写入硬盘**

**secure             NFS通过1024以下的安全TCP/IP端口发送   
 insecure          NFS通过1024以上的端口发送   
 wdelay            如果多个用户要写入NFS目录，则归组写入（默认）   
 no\_wdelay      如果多个用户要写入NFS目录，则立即写入，当使用async时，无需此设置。   
 Hide                在NFS共享目录中不共享其子目录   
 no\_hide           共享NFS目录的子目录   
 subtree\_check   如果共享/usr/bin之类的子目录时，强制NFS检查父目录的权限（默认）   
 no\_subtree\_check   和上面相对，不检查父目录权限   
 all\_squash               共享文件的UID和GID映射匿名用户anonymous，适合公用目录。   
 no\_all\_squash         保留共享文件的UID和GID（默认）   
 root\_squash             root用户的所有请求映射成如anonymous用户一样的权限（默认）   
 no\_root\_squash        root用户具有根目录的完全管理访问权限**

**[root@xue63 a]# cat /etc/exports**

**/tmp/a/no\_root\_squash      \*(rw,no\_root\_squash)**

**/tmp/a/sync               192.168.0.0/24(rw,sync)**

**/tmp/a/ro                  192.168.1.64(ro)**

**/tmp/a/all\_squash             192.168.0.0/24(rw,all\_squash,anonuid=500,anongid=500)**

**/tmp/a/async                    192.168.3.0/255.255.255.0(async)**

**/tmp/a/rw          192.168.3.0/255.255.255.0(rw)    192.168.4.0/255.255.255.0(rw)**

**/tmp/a/root\_squash   \*(rw,root\_squash)**

**注意：在发布共享目录的格式中除了共享目录是必跟参数外，其他参数都是可选的。并且共享**

**目录与客户端之间及客户端与客户端之间需要使用空格符号，但是客户端与参数之间是不能有**

**空格的**

**NFS客户端挂载参数的优化:**

**NFS高并发环境下的服务端重要优化(mount -o 参数）**

**async 异步同步，此参数会提高I/O性能，但会降低数据安全（除非对性能要求很高，对数据可靠性不要求的场合。一般生产环境，不推荐使用）**

**noatime 取消更新文件系统上的inode访问时间,提升I/O性能，优化I/O目的，推荐使用。**

**nodiratime 取消更新文件系统上的directory inode访问时间，高并发环境，推荐显式应用该选项，提高系统性能**

**intr：可以中断不成功的挂载**

**rsize/wsize 读取（rsize）/写入（wsize）的区块大小（block size），这个设置值可以影响客户端与服**

**务端传输数据的缓冲存储量。一般来说，如果在局域网内，并且客户端与服务端都具有足够的内存，这个**

**值可以设置大一点，比如说32768（bytes）,提升缓冲区块将可提升NFS文件系统的传输能力。但设置的值也不要太大，最好是实现网络能够传输的最大值为限。**

**内核优化：**

**net.core.wmem\_default = 8388608 #内核默认读缓存**

**net.core.rmem\_default = 8388608 #内核默认写缓存**

**net.core.rmem\_max = 16777216 #内核最大读缓存**

**net.core.wmem\_max = 16777216 #内核最大写缓存**

**用法:**

**mount -t nfs -o noatime,nodiratime,rsize=131072,wsize=131072,intr 192.168.0.63:/backup/NFS /mnt**

**或者写到挂载文件里:**

**192.168.0.63:/backup/NFS /mnt nfs noatime,nodiratime,rsize=131072,wsize=131072,intr 0 0**