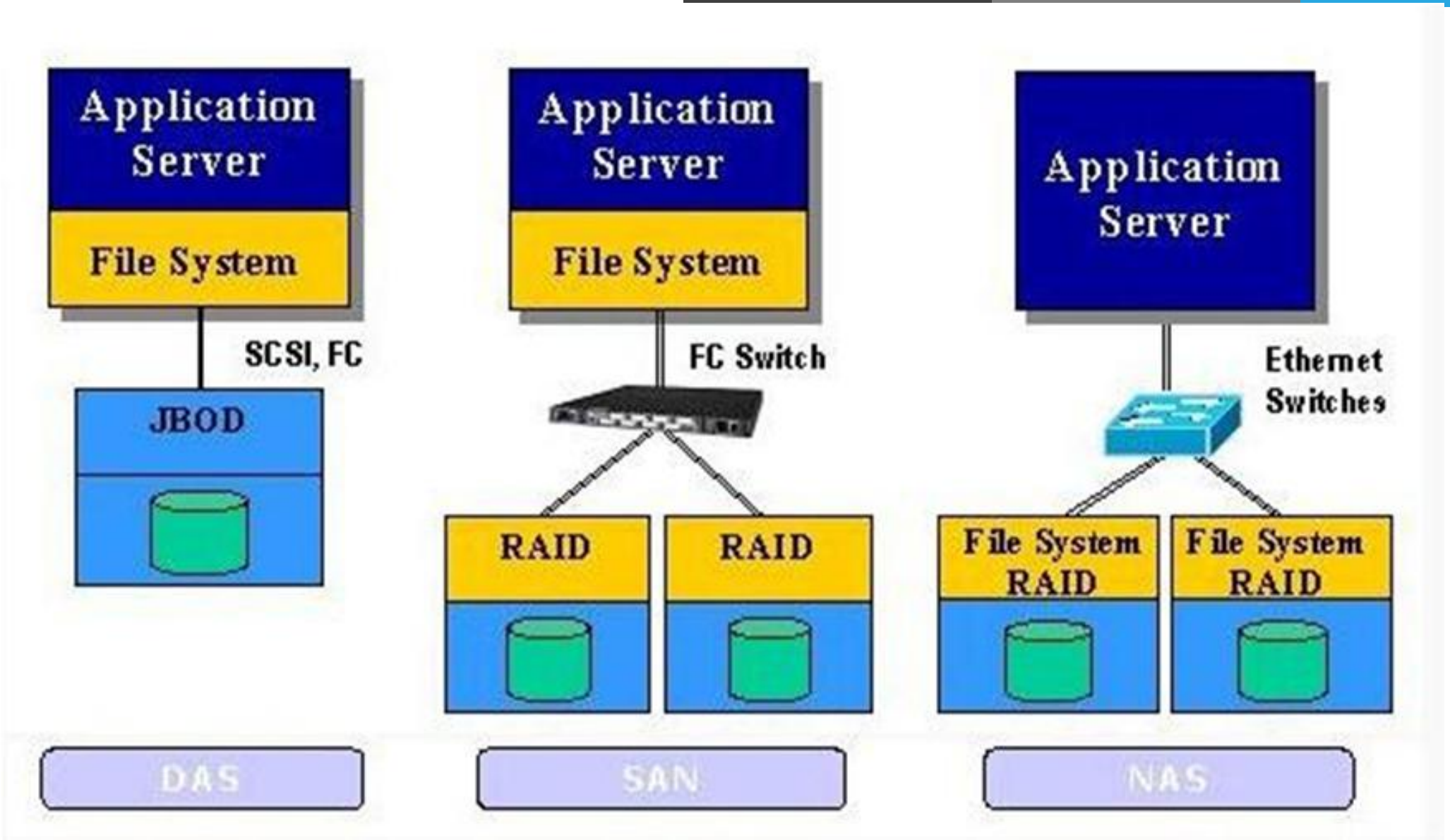
网络文件共享服务

讲师：王晓春

本章内容

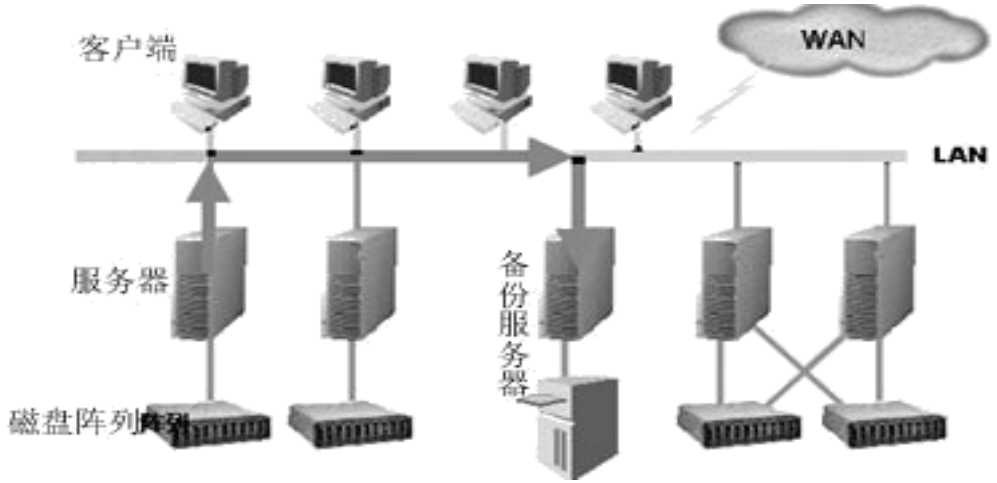
- FTP服务
- NFS服务
- SAMBA服务
- 数据同步

DAS、SAN 、NAS



存储基础知识---存储网络

直接存储(Direct Attached Storage)。存储设备与主机的紧密相连



- 管理成本较低，实施简单
- 储时直接依附在服务器上，因此存储共享受到限制
- CPU必须同时完成磁盘存取和应用运行的双重任务，所以不利于CPU的指令周期的优化，增加系统负担

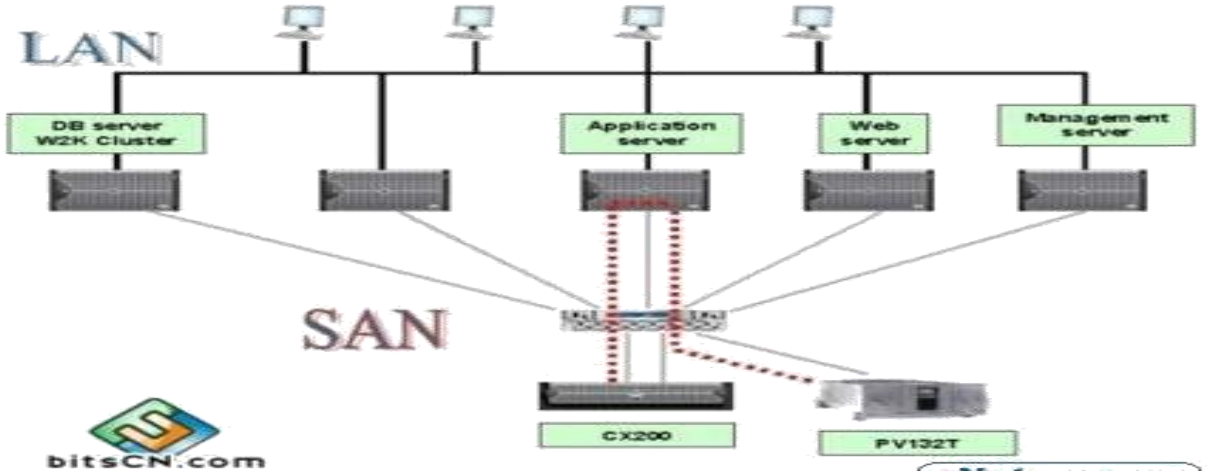
存储基础知识---存储网络

网络连接存储(Network Attached Storage)：
通过局域网在多个文件服务器之间实现了互联，基于文件的协议（ FTP、NFS、SMB/CIFS等 ），实现文件共享

- 集中管理数据，从而释放带宽、提高性能
- 可提供跨平台文件共享功能
- 可靠性较差，适用于局域网或较小的网络

存储基础知识---存储网络

存储区域网络(Storage Area Networks，SAN)
利用高速的光纤网络链接服务器与存储设备，基于SCSI，IP，ATM等多种高级协议，实现存储共享



- 服务器跟储存装置两者各司其职
- 利用光纤信道来传输数据，以达到一个服务器与储存装置之间多对多的高效能、高稳定度的存储环境
- 实施复杂，管理成本高

存储基础知识---存储网络

	DAS	NAS	SAN
传输类型	SCSI、FC	IP	IP、FC、SAS
数据类型	数据块	文件	数据块
典型应用	任何	文件服务器	数据库应用
优点	磁盘与服务器分离，便于统一管理	不占用应用服务器资源 广泛支持操作系统 扩展较容易 即插即用，安装简单方便	高扩展性 高可用性 数据集中，易管理
缺点	连接距离短 数据分散，共享困难 存储空间利用率不高 扩展性有限	不适合存储量大的块级应用 数据备份及恢复占用网络带宽	相比NAS成本较高 安装和升级比NAS复杂

文件传输协议FTP

- File Transfer Protocol 早期的三个应用级协议之一
- 基于C/S结构
- 双通道协议：数据和命令连接
- 数据传输格式：二进制（默认）和文本
- 两种模式：服务器角度

  主动(PORT style)：服务器主动连接

  命令（控制）：客户端：随机port --->服务器：tcp21

  数据：客户端：随机port <---服务器：tcp20

  被动(PASV style)：客户端主动连接

  命令（控制）：客户端：随机port --->服务器：tcp21

  数据：客户端：随机port --->服务器：随机port

- 服务器被动模式数据端口示例：

  227 Entering Passive Mode (172,16,0,1,224,59)

  服务器数据端口为：224*256+59

FTP软件介绍

- FTP服务器：
  - Wu-ftpd，Proftpd，Pureftpd，ServU，IIS
  - vsftpd:Very Secure FTP Daemon，CentOS默认FTP服务器
    - 高速，稳定，下载速度是WU-FTP的两倍
    - ftp.redhat.com数据:单机最多可支持15000个并发
- 客户端软件：
  - ftp，lftp，lftpget，wget，curl
    - ftp -A ftpserver port -A主动模式 –p 被动模式
    - lftp –u username ftpserver
    - lftp username@ftpserver
    - lftpget ftp://ftpserver/pub/file
  - gftp：GUI centos5 最新版2.0.19 (11/30/2008)
  - filezilla，CuteFtp，FlashFXP，LeapFtp
  - IE ftp://username:password@ftpserver

FTP服务

- 状态码：

  1XX：信息　　125：数据连接打开
  2XX：成功类状态　200：命令OK　230：登录成功
  3XX：补充类　　331：用户名OK
  4XX：客户端错误　425：不能打开数据连接
  5XX：服务器错误　530：不能登录

- 用户认证：

  匿名用户：ftp,anonymous,对应Linux用户ftp
  系统用户：Linux用户,用户/etc/passwd,密码/etc/shadow
  虚拟用户：特定服务的专用用户，独立的用户/密码文件
  　　nsswitch:network service switch名称解析框架
  　　pam:pluggable authentication module 用户认证
  　　/lib64/security　/etc/pam.d/　/etc/pam.conf

vsftpd服务

- 由vsftpd包提供
- 不再由xinetd管理
- 用户认证配置文件：/etc/pam.d/vsftpd
- 服务脚本： /usr/lib/systemd/system/vsftpd.service
  /etc/rc.d/init.d/vsftpd
- 配置文件：/etc/vsftpd/vsftpd.conf
  man 5 vsftpd.conf
  格式：option=value
  注意：= 前后不要有空格
- 匿名用户（映射为系统用户ftp ）共享文件位置：/var/ftp
- 系统用户共享文件位置：用户家目录
- 虚拟用户共享文件位置：为其映射的系统用户的家目录

vsftpd服务配置

- 命令端口
  - listen_port=21
- 主动模式端口
  - connect_from_port_20=YES　　主动模式端口为20
  - ftp_data_port=20（默认）　　指定主动模式的端口
- 被动模式端口范围
  - linux　　客户端默认使用被动模式
  - windows　　客户端默认使用主动模式
  - pasv_min_port=6000　　0为随机分配
  - pasv_max_port=6010
- 使用当地时间
  - use_localtime=YES　　使用当地时间（默认为NO，使用GMT）

vsftpd服务配置

- 匿名用户

anonymous_enable=YES　　支持匿名用户
no_anon_password=YES(默认NO)　　匿名用户略过口令检查
anon_world_readable_only (默认YES)只能下载全部读的文件
anon_upload_enable=YES　　匿名上传，注意:文件系统权限
anon_mkdir_write_enable=YES　匿名建目录
anon_umask=0333　　指定匿名上传文件的umask，默认077
anon_other_write_enable=YES　可删除和修改上传的文件

指定上传文件的默认的所有者和权限
chown_uploads=YES(默认NO)
chown_username=wang
chown_upload_mode=0644

vsftpd服务配置

- Linux系统用户

local_enable=YES	是否允许linux用户登录
write_enable=YES	允许linux用户上传文件
local_umask=022	指定系统用户上传文件的默认权限
guest_enable=YES	所有系统用户都映射成guest用户
guest_username=ftp	配合上面选项才生效，指定guest用户
local_root=/ftproot	guest用户登录所在目录

- 禁锢所有系统用户在家目录中

  chroot_local_user=YES（默认NO，不禁锢）禁锢系统用户

- 禁锢或不禁锢特定的系统用户在家目录中，与上面设置功能相反

  chroot_list_enable=YES

  chroot_list_file=/etc/vsftpd/chroot_list

  当chroot_local_user=YES时，则chroot_list中用户不禁锢

  当chroot_local_user=NO时，则chroot_list中用户禁锢

vsftpd服务配置

- wu-ftp日志：默认启用
  - xferlog_enable=YES （默认）　　启用记录上传下载日志
  - xferlog_std_format=YES （默认）　使用wu-ftp日志格式
  - xferlog_file=/var/log/xferlog （默认）可自动生成
- vsftpd日志：默认不启用
  - dual_log_enable=YES　　使用vsftpd日志格式，默认不启用
  - vsftpd_log_file=/var/log/vsftpd.log（默认）可自动生成
- 登录提示信息
  - ftpd_banner= “welcome to mage ftp server"
  - banner_file=/etc/vsftpd/ftpbanner.txt 优先上面项生效
- 目录访问提示信息
  - dirmessage_enable=YES (默认)
  - message_file=.message(默认)　　信息存放在指定目录下.message

vsftpd服务配置

- 使用pam(Pluggable Authentication Modules)完成用户认证
  - pam_service_name=vsftpd
    - pam配置文件:/etc/pam.d/vsftpd
    - /etc/vsftpd/ftpusers　　默认文件中用户拒绝登录
- 是否启用控制用户登录的列表文件
  - userlist_enable=YES　　默认有此设置
  - userlist_deny=YES(默认值)　　黑名单,不提示口令，NO为白名单
  - userlist_file=/etc/vsftpd/users_list 此为默认值
- vsftpd服务指定用户身份运行
  - nopriv_user=nobody (默认值)
- 连接数限制
  - max_clients=0　　最大并发连接数
  - max_per_ip=0　　每个IP同时发起的最大连接数

vsftpd服务配置

- 传输速率：字节/秒

anon_max_rate=0	匿名用户的最大传输速率
local_max_rate=0	本地用户的最大传输速率

- 连接时间：秒为单位

connect_timeout=60	主动模式数据连接超时时长
accept_timeout=60	被动模式数据连接超时时长
data_connection_timeout=300	数据连接无数据输超时时长
idle_session_timeout=60	无命令操作超时时长

- 优先以文本方式传输

ascii_upload_enable=YES

ascii_download_enable=YES

vsftpd服务配置

- 配置FTP服务以非独立服务方运行

vim /etc/vsftpd/vsftpd.conf/

listen=NO，默认为独立方式

vim /etc/xinetd.d/vsftpd

```
service ftp
{
        flags                   = REUSE
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/vsftpd
        log_on_failure          += USERID
        disable                 = no
}
```

实现基于SSL的FTPS

- 查看是否支持SSL
  - ldd `which vsftpd`　　查看到libssl.so
- 创建自签名证书
  - cd /etc/pki/tls/certs/
  - make vsftpd.pem
  - openssl x509 -in vsftpd.pem -noout –text
- 配置vsftpd服务支持SSL：/etc/vsftpd/vsftpd.conf
  - ssl_enable=YES　　启用SSL
  - allow_anon_ssl=NO　　匿名不支持SSL
  - force_local_logins_ssl=YES　本地用户登录加密
  - force_local_data_ssl=YES　本地用户数据传输加密
  - rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem
- 用filezilla等工具测试

vsftpd虚拟用户

- 虚拟用户：

  所有虚拟用户会统一映射为一个指定的系统帐号：访问共享位置，即为此系统帐号的家目录

  各虚拟用户可被赋予不同的访问权限，通过匿名用户的权限控制参数进行指定

- 虚拟用户帐号的存储方式：

  文件：编辑文本文件，此文件需要被编码为hash格式

  奇数行为用户名，偶数行为密码

  db_load -T -t hash -f vusers.txt vusers.db

  关系型数据库中的表中：

  实时查询数据库完成用户认证

  mysql库：pam要依赖于pam-mysql

  /lib64/security/pam_mysql.so

  /usr/share/doc/pam_mysql-0.7/README

实现基于文件验证的vsftpd虚拟用户

一、创建用户数据库文件

- vim /etc/vsftpd/vusers.txt

  wang

  wangpass

  mage

  magepass
- cd /etc/vsftpd/
- db_load -T -t hash -f vusers.txt vusers.db
- chmod 600 vusers.db

实现基于文件验证的vsftpd虚拟用户

二、创建用户和访问FTP目录

- useradd -d /var/ftproot -s /sbin/nologin vuser
- chmod +rx /var/ftproot/

- centos7 还需要执行以下操作：
- chmod -w /var/ftproot/
- mkdir /var/ftproot/upload
- setfacl -m u:vuser:rwx /var/ftproot/upload

实现基于文件验证的vsftpd虚拟用户

◆三、创建pam配置文件

- vim /etc/pam.d/vsftpd.db
  auth required pam_userdb.so db=/etc/vsftpd/vusers
  account required pam_userdb.so db=/etc/vsftpd/vusers

◆四、指定pam配置文件

- vim /etc/vsftpd/vsftpd.conf
  guest_enable=YES
  guest_username=vuser
  pam_service_name=vsftpd.db
- 五、SELinux设置：
  禁用SELinux 或者 setsebool -P ftpd_full_access 1

实现基于文件验证的vsftpd虚拟用户

- 六、虚拟用户建立独立的配置文件
- mdkir /etc/vsftpd/vusers.d/ 创建配置文件存放的路径
- vim /etc/vsftpd/vsftpd.conf
  user_config_dir=/etc/vsftpd/vusers.d/
- cd /etc/vsftpd/vusers.d/ 进入此目录
- 允许wang用户可读写，其它用户只读
- vim wang　创建各用户自已的配置文件
  anon_upload_enable=YES
  anon_mkdir_write_enable=YES
  anon_other_write_enable=YES
- vim mage 创建各用户自已的配置文件
  local_root=/ftproot 登录目录改变至指定的目录

实现基于MYSQL验证的vsftpd虚拟用户

- 说明：本实验在两台CentOS主机上实现，一台做为FTP服务器，一台做数据库服务器
- 一、安装所需要包和包组：
  - 在数据库服务器上安装包：
  - Centos7：在数据库服务器上安装
    yum –y install mariadb-server
    systemctl start mariadb.service
    systemctl enable mariadb
  - Centos6：在数据库服务器上安装
    yum –y install mysql-server
  - 在FTP服务器上安装vsftpd和pam_mysql包
    centos6：pam_mysql由epel6的源中提供
    yum install vsftpd pam_mysql

实现基于MYSQL验证的vsftpd虚拟用户

- centos7：无对应rpm包，需手动编译安装

```
yum -y groupinstall "Development Tools"
yum -y install mariadb-devel  pam-devel vsftpd
下载pam_mysql-0.7RC1.tar.gz
        ftp://172.16.0.1/pub/Sources/sources/pam/
tar xvf pam_mysql-0.7RC1.tar.gz
cd pam_mysql-0.7RC1/
./configure --with-pam-mods-dir=/lib64/security --with-mysql=/usr --with-pam=/usr

make
make install
```

实现基于MYSQL验证的vsftpd虚拟用户

二、在数据库服务器上创建虚拟用户账号

1.建立存储虚拟用户数据库和连接的数据库用户

```
mysql> CREATE DATABASE vsftpd;
mysql> SHOW DATABASES;
```

- ftp服务和mysql不在同一主机：

```
mysql> GRANT SELECT ON vsftpd.* TO
vsftpd@'172.16.%.%'  IDENTIFIED BY 'magedu';
```

- ftp服务和mysql在同一主机：

```
mysql> GRANT SELECT ON vsftpd.* TO
vsftpd@localhost IDENTIFIED BY 'magedu';
mysql> GRANT SELECT ON vsftpd.* TO
vsftpd@'127.0.0.1'  IDENTIFIED BY 'magedu';
mysql> FLUSH PRIVILEGES;
```

实现基于MYSQL验证的vsftpd虚拟用户

- 2.准备相关表

```
mysql> USE vsftpd;
Mysql> SHOW TABLES;
mysql> CREATE TABLE users (
id INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
name CHAR(50) BINARY NOT NULL,
password CHAR(48) BINARY NOT NULL
);
mysql>DESC users;
```

测试连接

```
mysql -uvsftpd -h mysqlserver  -pmagedu
mysql> SHOW DATABASES;
```

实现基于MYSQL验证的vsftpd虚拟用户

- 3.添加虚拟用户
- 根据需要添加所需要的用户，为了安全应该使用PASSWORD函数加密其密码后存储

```
mysql> DESC users;
mysql> INSERT INTO users(name,password) values(  ‘wang',password('magedu'));
mysql> INSERT INTO users(name,password) values(  ‘mage',password('magedu'));
mysql> SELECT * FROM users;
```

实现基于MYSQL验证的vsftpd虚拟用户

- 三、在FTP服务器上配置vsftpd服务
- 1.在FTP服务器上建立pam认证所需文件

vi /etc/pam.d/vsftpd.mysql 添加如下两行

auth required pam_mysql.so user=vsftpd passwd=magedu host=mysqlserver db=vsftpd table=users usercolumn=name passwdcolumn=password crypt=2

account required pam_mysql.so user=vsftpd passwd=magedu host=mysqlserver db=vsftpd table=users usercolumn=name passwdcolumn=password crypt=2

注意：参考README文档，选择正确的加密方式

crypt是加密方式，0表示不加密，1表示crypt(3)加密，2表示使用mysql password()函数加密，3表示md5加密，4表示sha1加密

配置字段说明

- auth 表示认证
- account 验证账号密码正常使用
- required 表示认证要通过
- pam_mysql.so模块是默认的相对路径，是相对/lib64/security/路径而言，也可以写绝对路径；后面为给此模块传递的参数
- user=vsftpd为登录mysql的用户
- passwd=magedu 登录mysql的的密码
- host=mysqlserver mysql服务器的主机名或ip地址
- db=vsftpd 指定连接msyql的数据库名称
- table=users 指定连接数据库中的表名
- usercolumn=name 当做用户名的字段
- passwdcolumn=password 当做用户名字段的密码
- crypt=2 密码的加密方式为mysql password()函数加密

实现基于MYSQL验证的vsftpd虚拟用户

- 2.建立相应用户和修改vsftpd配置文件，使其适应mysql认证
  建立虚拟用户映射的系统用户及对应的目录
  useradd -s /sbin/nologin -d /var/ftproot vuser
  chmod 555 /var/ftproot centos7 需除去ftp根目录的写权限
  mkdir /var/ftproot/{upload,pub}
  setfacl –m u:vuser:rwx /var/ftproot/upload
  确保/etc/vsftpd.conf中已经启用了以下选项
  anonymous_enable=YES
  添加下面两项
  guest_enable=YES
  guest_username=vuser
  修改下面一项，原系统用户无法登录
  pam_service_name=vsftpd.mysql

实现基于MYSQL验证的vsftpd虚拟用户

- 四、启动vsftpd服务
  service vsftpd start;systemctl start vsftpd
  chkconfig vsftpd on;systemctl enable vsftpd
  查看端口开启情况
  netstat -tnlp |grep :21
- 五、Selinux相关设置：在FTP服务器上执行
  - restorecon -R /lib64/security
  - setsebool -P ftpd_connect_db 1
  - setsebool -P ftp_home_dir 1
  - chcon -R -t public_content_rw_t /var/ftproot/
- 六、测试：利用FTP客户端工具,以虚拟用户登录验证结果
  - tail /var/log/secure

实现基于MYSQL验证的vsftpd虚拟用户

◆七、在FTP服务器上配置虚拟用户具有不同的访问权限

vsftpd可以在配置文件目录中为每个用户提供单独的配置文件以定义其ftp服务访问权限，每个虚拟用户的配置文件名同虚拟用户的用户名。配置文件目录可以是任意未使用目录，只需要在vsftpd.conf指定其路径及名称即可

- 1、配置vsftpd为虚拟用户使用配置文件目录

  vim /etc/vsftpd/vsftpd.conf

  添加如下选项

  user_config_dir=/etc/vsftpd/vusers_config

- 2、创建所需要目录，并为虚拟用户提供配置文件

  mkdir /etc/vsftpd/vusers_config/

  cd /etc/vsftpd/vusers_config/

  touch wang mage

实现基于MYSQL验证的vsftpd虚拟用户

- 3、配置虚拟用户的访问权限

虚拟用户对vsftpd服务的访问权限是通过匿名用户的相关指令进行的。如要让用户wang具有上传文件的权限，可修改/etc/vsftpd/vusers_config/wang文件，在里面添加如下选项并设置为YES即可,只读则设为NO

注意：需确保对应的映射用户对于文件系统有写权限

anon_upload_enable={YES|NO}

anon_mkdir_write_enable={YES|NO}

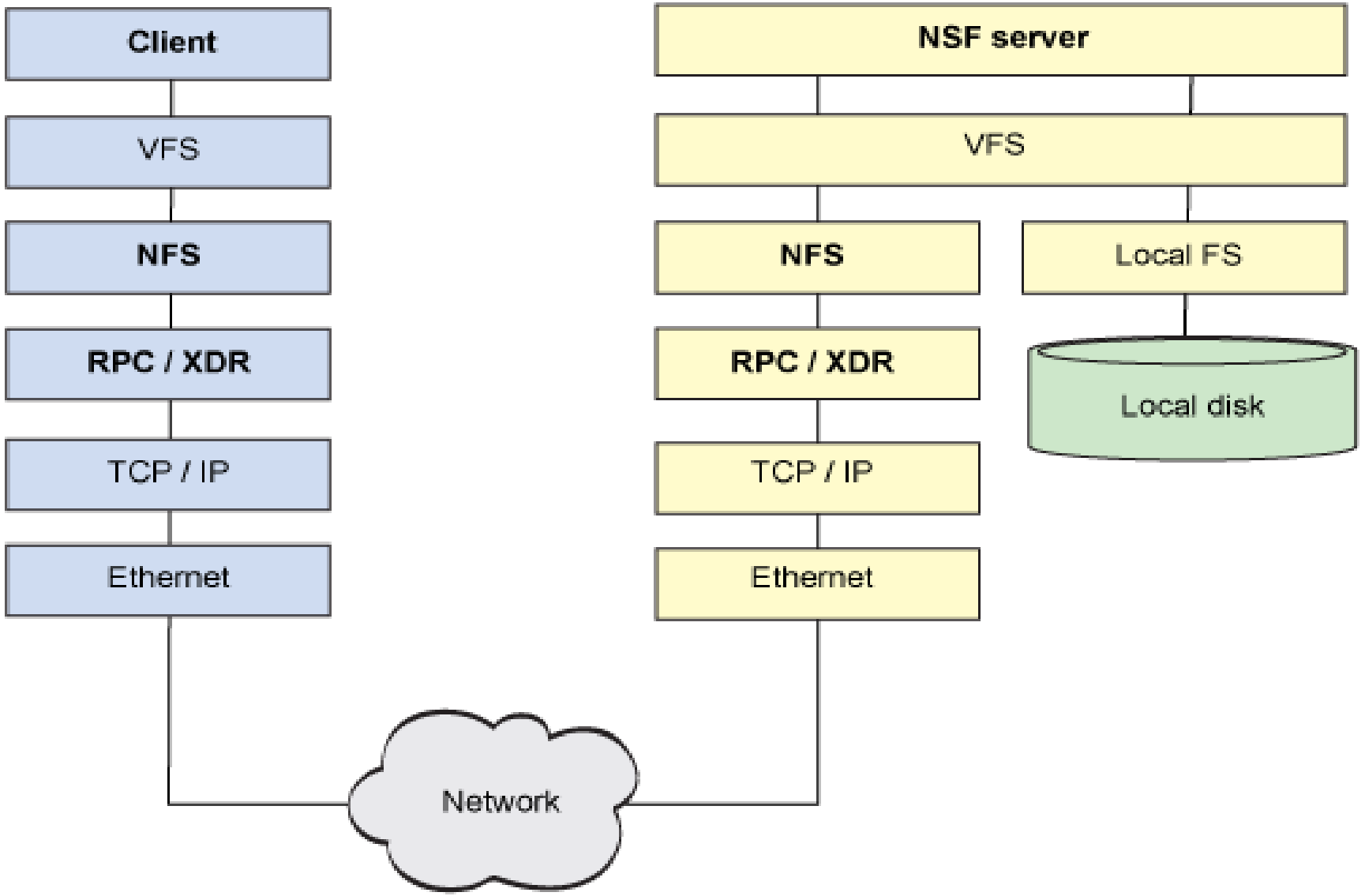
anon_other_write_enable={YES|NO}

local_root=/ftproot 登录目录改变至指定的目录

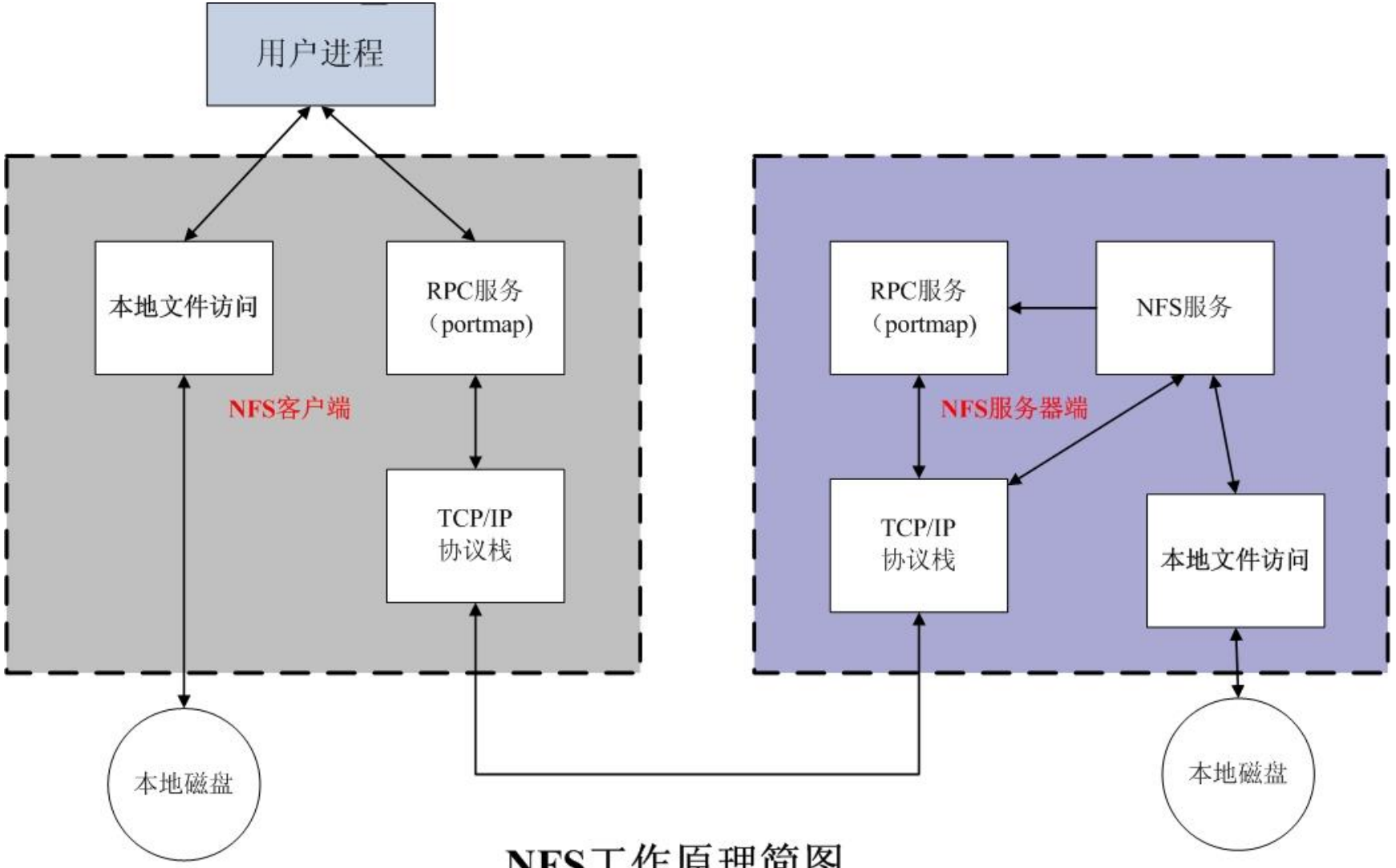
NFS服务

- NFS：Network File System 网络文件系统，基于内核的文件系统。Sun公司开发，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，基于RPC（Remote Procedure Call Protocol远程过程调用）实现
- RPC采用C/S模式。客户机请求程序调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行
- NFS优势：节省本地存储空间，将常用的数据,如home目录,存放在NFS服务器上且可以通过网络访问，本地终端将可减少自身存储空间的使用

NFS文件系统


Client
VFS
NFS
RPC / XDR
TCP / IP
Ethernet
Network
NSF server
VFS
NFS
Local FS
RPC / XDR
Local disk
TCP / IP
Ethernet

NFS工作原理



NFS工作原理简图

NFS各个版本的对比

NFS v2	NFS v3	NFS v4
只支持32位文件传输，最大文件数4G	支持64位文件传输	CentOS7默认很使用NFSv4版,实现伪根，辅助服务不需要，完全支持kerberos
文件传输尺寸限制在8K	没有文件尺寸限制	
无	V3增加和完善了许多错误和成功信息的返回,对于服务器的设置和管理能带来很大好处	改进了INTERNET上的存取和执行效能 在协议中增强了安全方面的特性
只提供了对UDP协议的支持,在一些高要求的网络环境中有很大限制	增加了对TCP传输协议的支持 有更好的I/O 写性能.	只支持TCP传输 通过一个安全的带内系统，协商在服务器和客户端之间使用的安全性类型 使用字符串而不是整数来表示用户和组标识符

NFS服务介绍

- 软件包：nfs-utils
- Kernel支持:nfs.ko
- 端口：2049(nfsd), 其它端口由portmap(111)分配
- 配置文件：/etc/exports,/etc/exports.d/*.exports
- CentOS7不支持同一目录同时用nfs和samba共享，因为使用锁机制不同
- 相关软件包:rpcbind（必须），tcp_wrappers
- CentOS6开始portmap进程由rpcbind代替
- NFS服务主要进程：

  rpc.nfsd　最主要的NFS进程，管理客户端是否可登录

  rpc.mountd 挂载和卸载NFS文件系统，包括权限管理

  rpc.lockd　非必要，管理文件锁，避免同时写出错

  rpc.statd　非必要，检查文件一致性，可修复文件
- 日志：/var/lib/nfs/

配置防火墙

- 配置防火墙，开放NFS服务
- 配置NFS使用固定端口
- vim /etc/sysconfig/nfs
  RQUOTAD_PORT=875
  LOCKD_TCPPORT=32803
  LOCKD_UDPPORT=32769
  MOUNTD_PORT=892
  STATD_PORT=662
  STATD_OUTGOING_PORT=2020
- 防火墙除开放上述端口，还需开放TCP和UDP的111和2049共4个端口

NFS配置文件

- 导出的文件系统的格式：

  /dir　　主机1(opt1,opt2)　主机2(opt1,opt2)...

- #开始为注释
- 主机格式：
  - 单个主机：ipv4，ipv6，FQDN
  - IP networks：两种掩码格式均支持

    172.18.0.0/255.255.0.0

    172.18.0.0/16
  - wildcards：主机名通配，例如*.magedu.com，IP不可以
  - netgroups：NIS域的主机组，@group_name
  - anonymous：表示使用*通配所有客户端

NFS配置文件

- 每个条目指定目录导出到的哪些主机，及相关的权限和选项
  - 默认选项：(ro,sync,root_squash,no_all_squash)
  - ro,rw 只读和读写
  - async 异步，数据变化后不立即写磁盘，性能高
  - sync（1.0.0后为默认）同步，数据在请求时立即写入共享
  - no_all_squash （默认）保留共享文件的UID和GID
  - all_squash 所有远程用户(包括root)都变成nfsnobody
  - root_squash （默认）远程root映射为nfsnobody,UID为65534，早期版本是4294967294 (nfsnobody)
  - no_root_squash 远程root映射成root用户
  - anonuid和anongid 指明匿名用户映射为特定用户UID和组GID，而非nfsnobody,可配合all_squash使用

NFS配置示例

- 在/etc/exports文件中定义导出目录
  - /myshare server.example.com
  - /myshare *.example.com
  - /myshare server?.example.com
  - /myshare server[0-20].example.com
  - /myshare 172.25.11.10
  - /myshare 172.25.0.0/16
  - /myshare 2000:472:18:b51:c32:a21
  - /myshare 2000:472:18:b51::/64
  - /myshare *.example.com 172.25.0.0/16
  - /myshare desktop.example.com(ro)
  - /myshare desktop.example.com(ro) server[0-20].example.com(rw)
  - /myshare diskless.example.com(rw,no_root_squash)

NFS工具

- rpcinfo
  - rpcinfo -p hostname
  - rpcinfo –s hostname 查看RPC注册程序
- exportfs
  - –v 查看本机所有NFS共享
  - –r 重读配置文件，并共享目录
  - –a 输出本机所有共享
  - –au 停止本机所有共享
- showmount -e hostname
- mount.nfs 挂载工具
- NFSv4支持通过挂载NFS服务器的共享“根”，从而浏览NFS服务器上的共享目录列表
  - mount nfsserver:/ /mnt/nfs

客户端NFS挂载

- 基于安全考虑，建议使用nosuid,nodev,noexec挂载选项
- NFS相关的挂载选项：

  fg（默认）前台挂载，bg后台挂载

  hard（默认）持续请求，soft 非持续请求

  intr 和hard配合，请求可中断

  rsize和wsize 一次读和写数据最大字节数，rsize=32768

  _netdev 无网络不挂载
- 示例：

  mount -o rw,nosuid,fg,hard,intr 172.16.0.1:/testdir /mnt/nfs/
- 开机挂载:/etc/fstab

  172.16.0.1:/public /mnt/nfs nfs defaults 0 0

自动挂载

- 可使用autofs按需要挂载NFS共享，在空闲时自动卸载
- 由autofs包提供
- 系统管理器指定由/etc/auto.master自动挂载器守护进程控制的挂载点
- 自动挂载监视器访问这些目录并按要求挂载文件系统
- 文件系统在失活的指定间隔5分钟后会自动卸载
- 为所有导出到网络中的NFS启用特殊匹配 -host 至“browse”
- 参看帮助：man 5 autofs
- 支持含通配符的目录名
  - *　　server:/export/&

直接匹配

- 直接匹配包括绝对路径名称
- 不会影响本地目录结构
- 示例：

```
/etc/auto.master:
/-                      /etc/auto.direct

/etc/auto.direct:
/foo                    server1:/export/foo
/user/local/            server1:/usr/local
```

实验：实现NFS服务

- systemctl start nfs-server
- systemctl enable nfs-server
- mkdir /nfsshare
- chown nfsnobody /nfsshare
- vi /etc/exports
  /nfsshare desktopX(rw)
- exporfs –r
- mkdir /mnt/nfsshare
- mount serverX:/nfsshare /mnt/nfsshare
- vim /etc/fstab
  nfsserver:/nfsshare /mnt/nfsshare nfs defaults 0 0
- mount -a

实验：实现NFS伪根

- 配置NFS服务器

```
vi /etc/fstab
/data/read    /exports/read none bind 0 0
/data2/write  /exports/write none bind 0 0
vi /etc/exports
/exports *(fsid=0,ro,crossmnt)
/exports/read 192.168.0.0/24(ro)
/exports/write 192.168.0.0/24(rw)
```

- 配置NFS客户端

```
mount nfsserver:/  /mnt/nfs
vi /etc/fstab
nfsserver:/ /mnt/ nfs4 ro 0 0
```

SAMBA服务简介

- SMB：Server Message Block服务器消息块，IBM发布，最早是DOS网络文件共享协议
- Cifs：common internet file system，微软基于SMB发布
- SAMBA:1991年Andrew Tridgell,实现windows和UNIX相通
- SAMBA的功能：
  - 共享文件和打印，实现在线编辑
  - 实现登录SAMBA用户的身份认证
  - 可以进行NetBIOS名称解析
  - 外围设备共享
- 计算机网络管理模式：
  - 工作组WORKGROUP：计算机对等关系，帐号信息各自管理
  - 域DOMAIN:C/S结构，帐号信息集中管理，DC,AD

SAMBA介绍

- 相关包：
  - Samba 提供smb服务
  - Samba-client 客户端软件
  - samba-common 通用软件
  - cifs-utils smb客户端工具
  - samba-winbind 和AD相关
- 相关服务进程：
  - smbd 提供smb（cifs）服务 TCP:139,445
  - nmbd NetBIOS名称解析 UDP:137,138
- 主配置文件：/etc/samba/smb.conf
  - 帮助参看：man smb.conf
- 语法检查： testparm [-v] [/etc/samba/smb.conf]
- 客户端工具：smbclient,mount.cifs

SAMBA服务器配置

- smb.conf继承了.ini文件的格式，用[ ] 分成不同的部分
- 全局设置：
  [global] 服务器通用或全局设置的部分
- 特定共享设置：
  [homes] 用户的家目录共享
  [printers] 定义打印机资源和服务
  [sharename] 自定义的共享目录配置
- 其中：#和;开头的语句为注释，大小写不敏感
- 宏定义：

%m 客户端主机的NetBIOS名	%M 客户端主机的FQDN
%H 当前用户家目录路径	%U 当前用户用户名
%g 当前用户所属组	%h samba服务器的主机名
%L samba服务器的NetBIOS名	%I 客户端主机的IP
%T 当前日期和时间	%S 可登录的用户名

SAMBA服务器全局配置

- workgroup 指定工作组名称
- server string 主机注释信息
- netbios name 指定NetBIOS名
- interfaces 指定服务侦听接口和IP
- hosts allow 可用“,” ，空格，或tab分隔，默认允许所有主机访问，也可在每个共享独立配置，如在[global]设置，将应用并覆盖所有共享设置

  IPv4 network/prefix: 172.25.0.0/24 IPv4前缀: 172.25.0.

  IPv4 network/netmask: 172.25.0.0/255.255.255.0

  主机名: desktop.example.com

  以example.com后缀的主机名: .example.com

  示例：

  ```
  hosts allow = 172.25.
  hosts allow = 172.25. .example.com
  ```

- hosts deny 拒绝指定主机访问

SAMBA服务器全局配置

- config file=/etc/samba/conf.d/%U 用户独立的配置文件
- Log file=/var/log/samba/log.%m 不同客户机采用不同日志
- log level = 2 日志级别，默认为0，不记录日志
- max log size=50 日志文件达到50K，将轮循rotate,单位KB
- Security三种认证方式：
  - share：匿名(CentOS7不再支持)
  - user：samba用户（采有linux用户，samba的独立口令）
  - domain：使用DC（DOMAIN CONTROLLER)认证
- passdb backend = tdbsam 密码数据库格式
- 实现samba用户：
  - 包： samba-common-tools
  - 工具：smbpasswd pdbedit
  - samba用户须是Linux用户，建议使用/sbin/nologin

管理SAMBA用户

- 添加samba用户

  smbpasswd -a <user>

  pdbedit -a -u <user>

- 修改用户密码

  smbpasswd <user>

- 删除用户和密码：

  smbpasswd –x <user>

  pdbedit –x –u <user>

- 查看samba用户列表：

  /var/lib/samba/private/passdb.tdb

  pdbedit –L –v

- 查看samba服务器状态

  smbstatus

配置目录共享

- 每个共享目录应该有独立的[ ]部分
  - [共享名称] 远程网络看到的共享名称
  - comment 注释信息
  - path 所共享的目录路径
  - public 能否被guest访问的共享，默认no，和guest ok 类似
  - browsable 是否允许所有用户浏览此共享,默认为yes,no为隐藏
  - writable=yes 可以被所有用户读写，默认为no
  - read only=no 和writable=yes等价，如与以上设置冲突，放在后面的设置生效，默认只读
  - write list 三种形式：用户，@组名，+组名,用，分隔

    如writable=no，列表中用户或组可读写，不在列表中用户只读
  - valid users 特定用户才能访问该共享，如为空，将允许所有用户，用户名之间用空格分隔

基于特定用户和组的共享

- 编辑/etc/samba/smb.conf

```
[share]
path = /app/dir
valid users=wang,@admins
writeable = no
browseable = no
```

SMB客户端访问

- UNC路径: Universal Naming Convention,通用命名规范
  格式：\\sambaserver\sharename
- 终端下使用smbclient登录服务器

smbclient -L instructor.example.com

smbclient -L instructor.example.com -U wang

> cd directory

> get file1

> put file2

smbclient //instructor.example.com/shared -U wang

可以使用-U选项来指定用户%密码，或通过设置和导出USER和PASSWD环境变量来指定

挂载CIFS文件系统

◆手动挂载

mount -t cifs -o user=wang,password=magedu //server//shared /mnt/smb

◆开机自动挂载

- cat /etc/fstab 可以用文件代替用户名和密码的输入

//server/homes /mnt cifs credentials=/etc/smb.txt 0 0

- cat /etc/smb.txt

  username=wang

  password=password
- chmod 600 /etc/smb.txt

实验：实现SMB共享

- 一、在samba服务器上安装samba包

  yum -y install samba

- 二、创建samba用户和组

  groupadd -r admins

  useradd -s /sbin/nologin -G admins wang

  smbpasswd -a wang

  useradd -s /sbin/nologin mage

  smbpasswd -a mage

实验：实现SMB共享

- 三、创建samba共享目录,并设置SElinux

```
mkdir /testdir/smbshare
chgrp admins /testdir/smbshare
chmod 2775 /testdir/smbshare

semanage fcontext -a -t samba_share_t '/testdir/smbshare(/.*)?'
restorecon -vvFR /testdir/smbshare
```

实验：实现SMB共享

- 三、samba服务器配置
  - vim /etc/samba/smb.conf

```
security = user
passdb backend = tdbsam
[share]
path = /testdir/smbshare
write list = @admins
```

  - systemctl start smb nmb
  - systemctl enable smb nmb
  - firewall-cmd --permanent --add-service=samba
  - firewall-cmd --reload

实验：实现SMB共享

- 四、samba客户端访问
  - 安装包

```
yum -y install cifs-utils
```

  - 用wang用户挂载smb共享并访问

```
mkdir /mnt/wang
mount -o username=wang //smbserver/share /mnt/wang
echo "Hello wang" >/mnt/wang/wangfile.txt
```

  - 用mage用户挂载smb共享并访问

```
mkdir /mnt/mage
mount -o username=mage //smbserver/share /mnt/mage
touch /mnt/mage/magefile.txt
```

多用户SMB挂载

- SAMBA共享默认只支持同时用一个用户挂载SMB共享
- CentOS7中可启用多用户挂载功能

客户端挂载samba共享目录后，在客户端登录的不同用户访问同一个samba的挂载点，可获得不同权限

实验：多用户SMB挂载

一、samba服务器配置

- yum install samba
- mkdir /multiuser
- vim /etc/samba/smb.conf

```
[smbshare]
path=/multiuser
writable=no
write list= @admins
```

实验：多用户SMB挂载

- 二、samba服务器创建samba用户

useradd –s /sbin/nologin smbuser
smbpasswd –a smbuser
useradd –s /sbin/nologin –G admins wang
smbpasswd –a wang
useradd –s /sbin/nologin mage
smbpasswd –a mage

实验：多用户SMB挂载

- 三、samba服务器设置目录权限和SELinux

  对wang,admins组分配目录读写权限

  chmod 777 /testdir/multiuser

  或者

  setfacl –m u:wang:rwx /testdir/multiuser

  setfacl –m g:admins:rwx /testdir/multiuser

  设置SELinux标签：

  semanage fcontext -a -t samba_share_t '/testdir/multiuser (/.*)?'

  restorecon /testdir/multiuser

实验：多用户SMB挂载

- 四、samba客户端启用多用户挂载

```
yum -y install cifs-utils
mkdir /mnt/smb
echo 'username=smbuser' >/etc/multiuser
echo 'password=centos' >>/etc/multiuser
chmod 600 /etc/multiuser
```

以多用户方式挂载：

```
vim /etc/fstab
 //smbserver/smbshare   /mnt/smb   cifs
 credentials=/etc/multiuser,multiuser  0 0
mount -a
```

实验：多用户SMB挂载

- 五、在samba客户端用实现多用户访问
  - useradd wang;useradd mage
  - 用root访问

    ls /mnt/smb;　　touch /mnt/smb/root.txt
  - 用wang访问

    ls /mnt/smb;　　touch /mnt/smb/wang.txt

    cifscreds add –u wang smbserver

    touch /mnt/smb/wang.txt
  - 用mage访问

    cifscreds add –u mage smbserver

    ls /mnt/smb

    touch /mnt/smb/mage.txt

数据的实时同步

- 实现实时同步
  - 要利用监控服务（inotify），监控同步数据服务器目录中信息的变化
  - 发现目录中数据产生变化，就利用rsync服务推送到备份服务器上
- 实现实时同步的方法
  - inotify+rsync 方式实现数据同步
  - sersync ：金山公司周洋在 inotify 软件基础上进行开发的，功能更加强大
- inotify：

异步的文件系统事件监控机制，利用事件驱动机制，而无须通过诸如cron等的轮询机制来获取事件，linux内核从2.6.13起支持 inotify，通过inotify可以监控文件系统中添加、删除，修改、移动等各种事件

- 实现inotify软件：

inotify-tools，sersync，lrsyncd

inotify和rsync实现实时同步

- inotify+rsync使用方式
  - inotify 对同步数据目录信息的监控
  - rsync 完成对数据的同步
  - 利用脚本进行结合

inotify和rsync实现实时同步

- 查看服务器内核是否支持inotify
  - Linux下支持inotify的内核最小为2.6.13
  - ll /proc/sys/fs/inotify　#列出下面的文件，说明服务器内核支持inotify
    -rw-r--r-- 1 root root 0 Dec 7 10:10 max_queued_events
    -rw-r--r-- 1 root root 0 Dec 7 10:10 max_user_instances
    -rw-r--r-- 1 root root 0 Dec 6 05:54 max_user_watches
- inotify内核参数
  - 参数说明：参看man 7 inotify
    max_queued_events：inotify事件队列最大长度，如值太小会出现 Event Queue Overflow 错误，默认值：16384
    max_user_watches：可以监视的文件数量（单进程），默认值：8192
    max_user_instances：每个用户创建inotify实例最大值，默认值：128

inotify和rsync实现实时同步

- inotify参考文档
  - https://github.com/rvoicilas/inotify-tools/wiki
- 安装：基于epel源

  yum install inotify-tools
- Inotify-tools包主要文件：
  - inotifywait： 在被监控的文件或目录上等待特定文件系统事件（open close delete等）发生，常用于实时同步的目录监控
  - inotifywatch：收集被监控的文件系统使用的统计数据，指文件系统事件发生的次数统计

inotify和rsync实现实时同步

- inotifywait命令常见选项
  - -m, --monitor　始终保持事件监听
  - -d, --daemon　以守护进程方式执行，和-m相似，配合-o使用
  - -r, --recursive　递归监控目录数据信息变化
  - -q, --quiet　输出少量事件信息
  - --timefmt <fmt> 指定时间输出格式
  - --format <fmt>　指定的输出格式；即实际监控输出内容
  - -e　指定监听指定的事件，如果省略，表示所有事件都进行监听
  - --exclude <pattern>　指定排除文件或目录，使用扩展的正则表达式匹配的模式实现
  - --excludei <pattern> 和exclude相似，不区分大小写
  - -o, --outfile <file>打印事件到文件中，相当于标准正确输出
  - -s, --syslogOutput 发送错误到syslog相当于标准错误输出

inotify和rsync实现实时同步

- --timefmt <fmt>时间格式，参考 man 3 strftime
  - %Y 年份信息，包含世纪信息
  - %y 年份信息，不包括世纪信息
  - %m 显示月份，范围 01-12
  - %d 每月的第几天，范围是 01-31
  - %H 小时信息，使用 24小时制，范围 00-23
  - %M 分钟，范围 00-59
- 示例：

```
--timefmt "%Y-%m-%d %H:%M"
```

inotify和rsync实现实时同步

- --format <fmt> 格式定义
  - %T 输出时间格式中定义的时间格式信息，通过 --timefmt option 语法格式指定时间信息
  - %w 事件出现时，监控文件或目录的名称信息
  - %f 事件出现时，将显示监控目录下触发事件的文件或目录信息，否则为空
  - %e 显示发生的事件信息，不同的事件默认用逗号分隔
  - %Xe显示发生的事件信息，不同的事件指定用X进行分隔
- 示例：

```
--format "%T %w%f event: %;e"
--format '%T  %w  %f'
```

inotify和rsync实现实时同步

- -e 选项指定的事件类型
  - create　　文件或目录创建
  - delete　　文件或目录被删除
  - modify　　文件或目录内容被写入
  - attrib　　文件或目录属性改变
  - close_write　　文件或目录关闭，在写入模式打开之后关闭的
  - close_nowrite　　文件或目录关闭，在只读模式打开之后关闭的
  - close　　文件或目录关闭，不管读或是写模式
  - open　　文件或目录被打开
  - moved_to　　文件或目录被移动到监控的目录中
  - moved_from　　文件或目录从监控的目录中被移动
  - move　　文件或目录不管移动到或是移出监控目录都触发事件
  - access　　文件或目录内容被读取
  - delete_self　　文件或目录被删除，目录本身被删除
  - unmount　　取消挂载
- 示例： -e create,delete,moved_to,close_write

inotify和rsync实现实时同步

- 示例：
- 监控一次性事件

  inotifywait /data
- 持续监控

  inotifywait -mrq /data
- 持续后台监控，并记录日志

  inotifywait -o /root/inotify.log -drq /data --timefmt "%Y-%m-%d %H:%M" --format "%T %w%f event: %e"
- 持续后台监控特定事件

  inotifywait -mrq /data --timefmt "%F %H:%M" --format "%T %w%f event: %;e" -e create,delete,moved_to,close_write

inotify和rsync实现实时同步

- 配置 rsync 服务器端的配置文件

```
vi /etc/rsyncd.conf
uid = root
gid = root
use chroot = no
max connections = 0      #最大连接数不限制
ignore errors    #
exclude = lost+found/
log file = /var/log/rsyncd.log #
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsyncd.lock
reverse lookup = no
hosts allow = 192.168.8.0/24
[backup]
path = /backup/
comment = backup
read only = no  #
auth users = rsyncuser #
secrets file = /etc/rsync.pass #
```

inotify和rsync实现实时同步

- 服务器端生成验证文件

```
echo "rsyncuser:magedu" > /etc/rsync.pass
chmod 600 /etc/rsync.pass
```

- 服务器端准备目录

```
mkdir /backup
```

- 服务器端启动rsync服务

```
rsync --daemon      可加入/etc/rc.d/rc.local实现开机启动
```

- 客户端配置密码文件

```
echo "magedu"  > /etc/rsync.pass
chmod 600 /etc/rsync.pass
```

- 客户端测试同步数据

```
rsync -avz --password-file=/etc/rsync.pass /data/ rsyncuser@rsync服务器IP::backup
```

inotify和rsync实现实时同步

- 创建inotify_rsync.sh脚本

```
#!/bin/bash
SRC='/data/'
DEST='rsyncuser@rsync服务器IP::backup'
inotifywait  -mrq --timefmt '%Y-%m-%d %H:%M' --format '%T %w %f' -e create,delete,moved_to,close_write ${SRC} |while read DATE TIME DIR FILE;do
        FILEPATH=${DIR}${FILE}
        rsync -az --delete  --password-file=/etc/rsync.pass $SRC $DEST && echo "At ${TIME} on ${DATE}, file $FILEPATH was backuped up via rsync" >> /var/log/changelist.log
done
```

关于马哥教育

马哥教育
IT人的高薪职业学院

- 博客：http://mageedu.blog.51cto.com
- 主页：http://www.magedu.com
- QQ：1661815153, 113228115
- QQ群：203585050, 279599283

祝大家学业有成

谢　谢