服务

nfs

重点 yum install rpcbind nfs-utils
共享目录的权限
showmount -e nfsIP 查看服务器 的挂载
mount nfsIP:/nfs /mnt 挂载在本地的mnt目录
no_root_squash 保留root的权限 不映射
all_squash,anonuid=1000,anongid=1000) //所有人都映射为匿名用户（包括root），匿名用户设置为UID＝1000的那个用户

NFS network file system 简单的文件按共享 unix -linux 之间的文件共享 c/s

触发挂载 autofs 创建触发点
本地触发挂载
[root@365linux ~]# vim /etc/auto.master
#/misc /etc/auto.misc
/mnt /etc/auto.misc
挂载点

[root@365linux ~]# vim /etc/auto.misc
nfs -fstype=nfs 192.168.122.109:/public
挂在的类型 挂载的目录

服务的配置和应用过程： 装 配 启 测 sync 同步
rsync 异步
装： 安装
配： 配置
启： 启动
测： 测试

1.NFS的安装

首先检查rpcbind和nfs-utils有没有安装，没有就装上。

[root@geust02 ~]# rpm -qa |grep rpcbind

[root@geust02 ~]# rpm -qa |grep nfs-utils

[root@geust02 ~]# yum install -y rpcbind nfs-utils

rpcbind ： 端口管理（nfs提供服务时，使用了一些随机端口，客户端在连接时，先要向rpcbind询问nfs的工作端口）
nfs-utils： nfs的配置工具（nfs提供文件共享的功能是在内核当中实现,所以他的主配置文件是/etc/export）

++++++++++++++++++nfs的共享功能是Linux内核默认就提供。 +++++++++

2. NFS配置共享某个目录
vim /etc/exports

/public *(ro,sync)
共享的目录 所有主机只读同步
[root@geust02 ~]# mkdir /public

3. NFS的启动
[root@geust02 ~]# systemctl start rpcbind
[root@geust02 ~]# systemctl enable rpcbind

[root@geust02 ~]# systemctl start nfs-server
[root@geust02 ~]# systemctl enable nfs-server

[root@geust02 ~]# rpcinfo -p 查看rpc支持的服务
启动NFS时要首先启动rpcbind。NFS启动时将自己的随机端口向rpcbind进行注册。

4. 客户端测试
[root@365linux ~]# showmount -e 192.168.122.109 查看挂载 ip的搭建nfs服务的ip
Export list for 192.168.122.109:
/public *

[root@365linux ~]# mount 192.168.122.109:/public /mnt 把nfs服务器的共享目录挂在到本地

/public *(rw,sync)
NFS支持写入

[root@geust02 ~]# exportfs -av 不停止rpcbind和nfs 的情况下重启nfs服务
exporting *:/public
使配置生效

吧nfs服务器的的共享目录挂在到本地是，用户是没有执行权限的 。需要在服务器段给文件夹中的其他人权限

1. 实现NFS共享文件夹可写，必须要满足nfs支持rw的选项以及共享的文件夹/public有写入权限（nfs可写和文件夹可写同时成立）。
2. 默认情况下， 客户端root的权限被映射成为匿名用户nfsnobody.

NFS的应用实例2： NFS客户端和服务器端的用户映射

结论：
1. NFS默认保留普通用户的文件拥有者的身份，但是，Linux系统对于用户的识别是通过UID来完成的，有可能造成，在客户端和服务器，同一个UID对应的用户名不一样。比如在客户端1001对应的lisi用户，而在服务端对应是zhangsan用户。要注意这点。
2. 而管理员root默认会被映射为nfsnobody，可以通过配置，可以取消root用户的匿名映射。
3. 可以通过配置，使任何普通用户的访问映射成某个匿名用户，

示例：
/public *(rw,sync,no_root_squash) //不把root映射为nfsnobody （保留root的权限）
/public *(rw,sync,all_squash,anonuid=1000,anongid=1000) //所有人都映射为匿名用户（包括root），匿名用户设置为UID＝1000的那个用户

NFS主要用在局域网内，提供服务器的后端数据存储，对于连接限制应该在物理连接上隔离。所以nfs服务本身的对主机名或IP的访问控制应用的意义不大

NFS的应用实例4： NFS客户端自动挂载　可以把挂载信息写在/etc/fstab中
[root@365linux ~]# vim /etc/fstab
192.168.122.109:/public　/mnt　　nfs　　defaults　　0 0

实现方式2： autofs　（触发挂载）

[root@365linux ~]# rpm　-qa　|grep　autofs
autofs-5.0.7-54.el7.x86_64

配： 默认配置即可使用
启：
[root@365linux ~]# systemctl　start　autofs
autofs实现了，当你去访问服务器的共享时，会自动把服务共享文件夹挂载到本地的/net/IP/共享文件夹。当你长时间不访问，它就自动卸载。效果如下：
[root@365linux ~]# ls　/net
[root@365linux ~]# cd　/net/192.168.122.109/
[root@365linux 192.168.122.109]# ls
autofs实现的第二种方式，可以自定义本地挂载点。

[root@365linux ~]# vim /etc/auto.master
#/misc　/etc/auto.misc
/mnt　　/etc/auto.misc

[root@365linux ~]# vim　/etc/auto.misc
nfs　　-fstype=nfs　　192.168.122.109:/public
服务　　服务类型　　serverip　　共享目录
[root@365linux ~]# systemctl　restart　autofs

客户端测试效果：
[root@365linux ~]# ls /mnt
[root@365linux ~]# cd　/mnt/nfs
[root@365linux nfs]# ls
demo.txt　ls.txt　　root.txt　　write.test
lisi.txt　man_db.conf　wangwu.txt　zhangsan.txt
[root@365linux nfs]# df　-h　|grep　mnt
192.168.122.109:/public　18G　1.8G　16G　11% /mnt/nfs

PS：autofs自动卸载的的超时时间设置：
[root@xueing nfs]# vim /etc/sysconfig/autofs
TIMEOUT=300

自动挂载aotufs 2
　vim /etc/auto.master　　定义挂载点
/nfs　　/etc/auto.nfs

cp /etc/auto.misc　　/etc/auto.nfs　　拷贝模板

vim /etc/auto.nfs　　编辑模板
user01　　-rw,soft,intr　　172.25.0.10:/nfs/share1
user02　　-rw,soft,intr　　172.25.0.10:/nfs/share2
user03　　-rw,soft,intr　　172.25.0.10:/nfs/share3
　　读写　软链接　网络
service autofs restart

nfs 触发挂载 如果出触发点的名字和 名字相同 可以 & * 匹配

```
[root@rhel7 ~]# rpm -ql nfs-utils
/etc/exports.d                              ----扩展配置目录
/etc/sysconfig/nfs                          ----额外配置文件
/sbin/mount.nfs                             ----二进制命令
/sbin/mount.nfs4
/sbin/osd_login
/sbin/rpc.statd
/sbin/umount.nfs
/sbin/umount.nfs4
/usr/lib/systemd/system/nfs.service  ----启动脚本
/usr/sbin/showmount                             ----查看共享目录
```

nfs作业

——

练习：
1. NFS的共享目录/nfs/public ， 实现所有用户可写，并权限映射到zhangsan.
2. NFS的共享目录/nfs/data, 实现root用户可写，其他用户只读，并不映射为匿名用户。
3. 配置客户端使用fstab实现重启后自动挂载nfs的data共享到/data目录。
4. 使用autofs实现访问时自动挂载nfs的public共享到/nfs/pub目录。

补充： NFS配置文件的帮助文档
]# man exports

script

写一个脚本自动搭建nfs服务

```
#!/bin/bash

#1.安装软件
#2.确认软件是否安装
#3.配置
#(1).新建共享目录,授本地权限
#(2).发布共享目录/etc/exports
#4.启动服务
#5.设置下次开机自动启动

#配置网络,测试网络

ping -c 1 192.168.0.254 &> /dev/null  && echo "###############网络OK###############"

#配置network yum

rm -fr /etc/yum.repos.d/*
cat > /etc/yum.repos.d/dvd.repo  << EOT
[base]
baseurl=ftp://192.168.0.254/rhel6_dvd
gpgcheck=0
EOT

#1.安装软件
yum -y install nfs* rpcbind  &> /dev/null  && echo "###############软件安装
OK###############"

#2.xx

#3.配置
#(1).新建共享目录,授本地权限

#read -p "请输入你的共享目录" dir
mkdir -p $dir
chmod 777 $dir && echo "###############本地授权OK###############"


#(2).发布共享目录/etc/exports

#read -p "请输共享主机地址和权限(192.168.0.0/24(xx)):" HOST

cat >> /etc/exports << EOT
$1 $2
EOT


#4.启动服务
service rpcbind restart &>/dev/null && echo "############rpcbind启动成功#############"
service nfs restart &>/dev/null && echo "################nfs启动成功################"


#5.设置下次开机自动启动

chkconfig rpcbind on
chkconfig nfs on
```

dhcp

dhcp （Dynamic host configuration protocol） 动态主机配置协议 是一个局域网的网络协议

UDP协议 67 服务器端使用
68 客户端使用

原理

请求IP租约 （DHCPDISCOVER包）---.> 提供租约（ DHCPOFFER-）--→ 选择租约
DHCPREQUEST ---->确认IP租约DHCPACK
得到IP后，客户端会发送一个ARP 请求来避免由于DHCP服务器地址次重启而引发的IP 冲突

一 给DHCP服务器配置一个静态IP地址

装

yum install - y dhcp

配 (一般情况下如果服务主配置文件没内容 那么在/usr/share/doc/下都能找到模板)
拷贝dhcp住配置模板
cp /usr/share/doc/dhcp-4.2.5/dhcpd.conf.example /etc/dhcp/dhcpd.conf

vim /etc/dhcp/dhcd.conf

```
option domain-name "example.org";                                  ----域名
option domain-name-servers ns1.example.org, ns2.example.org;       ----指定DNS 如果下面
subnet 没有指定 则 客户端的DNS 是这个

default-lease-time 600;                                            ----最小租约时间
max-lease-time 7200;                                                   ----最大租约时间

log-facility local7;                                                       ----日志

subnet 192.168.10.0 netmask 255.255.255.0 {                        ----局部配置文件
  range 192.168.10.10 192.168.10.20;                               ----地址池
  option domain-name-servers 192.168.10.1;                         ----授予客户端的DNS服务
器ip/hostname
  option domain-name "example.com";                                ----客户端接受域名
  option routers 192.168.10.1;                                         ----客户端接受网关
 # option broadcast-address 10.5.5.31;                                ----广播地址
  default-lease-time 600;                                          ----最小租约
  max-lease-time 7200;                                                 ----最大租约
}
```

subnet ----发布网段
netmask ----子网掩码

ifconfig eth0 192.168.10.1 ----将网卡ip地址修改成为网关

启

systemctl start dhcpd

测试

1、ip地址　　ifconfig
2、网关　　route -n
3、dns/domain　　cat /etc/resolv.conf

dhclient　-v　显示获取的过程

练习

DHCP的练习要求：

DHCP的服务器为分别处于两个网络的客户端分配IP
一个属于privnet01网络， 分配得到192.168.100.x/24的ip
一个属于privnet02网络， 该机器固定得到172.16.100.100/24的IP
客户端通过dhclient进行获取IP的测试。

script

注意事项　变量要用{} 括起来

```
#!/bin/bash
##这是批量部署DHCP的脚本
###2017/8/17
###yum
rm -rf /etc/yum.repos.d/*
cat > /etc/yum.repos.d/server.repo<< END
[server]
name=rhel7.iso
baseurl=http://172.25.254.250/rhel7.2/x86_64/dvd/
enabled=1
gpgcheck=0
END
        yum clean all &>/dev/null
        systemctl  stop firewalld
        systemctl  disable firewalld  &>/dev/null
        setenforce 0  &>/dev/null
        sed  -i  '/SELINUX=enforcing/c SELINUX=disabled' /etc/selinux/config
####装
    rpm -q dhcp
        if [ $? -ne 0  ];then
                        yum install -y dhcp &>/dev/null
                                if [ $? -ne 0 ];then
                                        echo "软件安装失败"
```

```
                                        else
                                                echo "软件已安装"
                                        fi
                        fi
###配置
read -p "请输入DNS服务器的IP :" D1
read -p  "请输入DHCP分配的网段如（192.168.3.0）： " D2
read -p  "请输入DHCP分配的网段的掩码如（255.255.255.0 ： " M1
read -p  "请输入你要分配的地址范围如（192.168.3.10 ）： " R1
read -p  "请输入你要分配的地址范围如（192.168.3.100）： " R2
read -p  "请输入你分配的地址的广播地址（192.168.3.255）： " B2
read -p  "请输入你的网关如（192.168.3.254）： " G1
cat >/etc/dhcp/dhcpd.conf <<EOF
option domain-name "example.org";
option domain-name-servers ${D1};
default-lease-time 600;
max-lease-time 7200;
log-facility local7;
subnet ${D2} netmask ${M1} {
  range ${R1} ${R2};
  option domain-name-servers ${D1};
  option domain-name "internal.example.org";
  option routers ${G1};
  option broadcast-address ${B2};
  default-lease-time 600;
  max-lease-time 7200;
}
#host fantasia {
#  hardware ethernet  00:0c:29:25:19:a3;
#  fixed-address 192.168.3.2;
#}
EOF


######启动
            systemctl  restart dhcpd
            [ $? -eq 0 ]  &&  echo "服务配置成功"
```

dhcp error

如果配置文件没有错重启服务的时候没有提示那行写错了
        1.本机没有IP静态地址　或者　 dhcp的地址池和本机不是同一个网段的

samba

网络文件系统之二 samba (cifs)　协议 smb/cifs

主要是用来linux跟windows之间共享数据，Linux和Linux之间也可以。主要用于局域网内。
smbd 139 445 TCP 文件传输
nmbd 137 138 UDP netbios 域名解析
装－ －》配－－》启－－》测
1. 安装
[root@geust02 ~]# rpm -qa |grep samba
[root@geust02 ~]# yum install samba

配置
[root@geust02 ~]# vim /etc/samba/smb.conf

security = user ---用户验证
map to guest = Bad User ---开启匿名访问
passdb backend = tdbsam

[public] ----共享目录的名称
comment = Public Stuff ----共享目录的解释说明
path = /samba ----共享目录在本地路径
public = yes/no ----是否为公共目录
writable = yes ----允许所有访问的用户写入
write list = +staff ---写入用户列表 只有这里面的用户有写入的权限
read list =adsf ----只读用户

启
systemctl start smb

测
在windows下面访问：在文件浏览器里面，输入\\192.168.122.109
在Linux下，
1.链接到服务器，输入smb://192.168.122.109
2.在Nautilus的地址栏里面输入smb://192.168.122.109
3.命令行下面访问samba
需要安装 samba-client， cifs-utils
登陆访问
[root@vhost01 ~]# smbclient -L 192.168.122.109 可以 -U user01 指定用户user01

挂载到本地访问
[root@teacher01 ~]# mount -t cifs //192.168.122.109/pub /mnt -o user=zhangsan 指定用户张三
或者：
[root@teacher01 ~]# mount //192.168.122.109/pub /mn

例子

1、修改samba配置文件
[财务部门]
comment= this is cw dep file
path = /uplooking/cw
public = no
valid users = cw01,cw02,boss01
write list = cw01,boss01
read list = @cw,boss01
2. 创建目录
mkdir /uplooking/cw
3.创建用户 和组

groupadd uplooking
groupadd cw
user add -g wc -G uplooking -s /sbin/nologin cw 01 smb使用的系统的用户，但是使用的用户密码是samba自己管理的密码，而非系统密码

（将存在的系统用户添加到samba自己用户认证体系） smb的用户和客户端没有关系

smbpasswd -a cw01 添加用户为smb用户
pdbedit -L 列出已添加的smb用户

++重要的一匹++++++++++++++++++++++++++++
4 对目录进行授权
chmod 750 /uplooking -R 更改目录的权限
chgrp uplooking /uplooking 更改文件夹的属组
chown cw01.cw /uplooking/cw 更改文件夹的拥有者和属组
++

访问控制ACL
setfacl -m u:boos01:rwx /uplooking/cw -m 修改 给指定用户设置 指定文件的权限

getfacl ---查看

粘滞位： 针对公共目录 sticky
在公共目录中，用户只能管理(删除)自己的文件（拥有者）
chmod 777 /uplooking/pub
chmod o+t /uplooking/pub
或者
chmod 1777 /uplooking/pub

作业

扩展（smb+lvm+quota+acl）

要求：
公司： www.uplooking.org
部门: cw rs sc

1.财务部门，只有财务总监（cw01）可以修改文件，财务成员(cw02)审核文件，boss01 汇总公司财务
2.人事部门，人事部门可以修改文件，公司所有员工都可以查看文件，boss02 也可以修改文件
3.市场部门，只有市场总监可以修改文件，市场成员可以查询，boss03 也可以修改文件
4.vip 可以访问rs，sc
5.要求所有boss对目录只能写入100M，创建文件不能超过200个
6.在公共目录自己文件自己管理

问题

搞懂权限就没问题了
windows 没有权限

大目录给个777
有拥有者就给750

net use * /delete

linux 无法挂载到本地

apache script

作业1：
写一个自动搭建apache服务的脚本，要求如下：
1、用户输入web服务器的IP、域名以及数据根目录
2、如果用户不输入则一直提示输入，直到输入为止
3、当访问www.test.cc时可以访问到数据根目录里的首页文件“this is test page”
搭建基于域名的虚拟主机：
1> 关闭防火墙和selinux
2> 配置yum源（本地|内网）
3> 软件三步曲（查看软件是否安装|安装（确定是否成功安装）|查看软件列表）
4> 了解配置文件 man 5 xxx.conf
5> 根据需求通过修改配置文件来完成服务的搭建
6> 启动服务|开机自启动
7> 测试验证

```
#!/bin/bash
conf=/etc/httpd/conf/httpd.conf
fun_web(){
input=""
output="$1"
while [ -z $input ]
  do
          read -p "$output" input
  done
echo $input
}

#获取用户所输入的ip、hostanme、root_dir并赋值给变量
ip=$(fun_web "请输入你的IP地址(10.1.1.1)：")
hostname=$(fun_web "请输入你的主机名(www.test.cc)：")
root_dir=$(fun_web "请输入你的数据根目录(/var/www/html)：")

#修改hosts文件将域名和ip对应起来
cat >>/etc/hosts<<end
$ip $hostname
end

#判断数据根目录是否存在并创建首页文件及修改权限
```

```
[ ! -d $root_dir ] && mkdir -p $root_dir
echo "this is test page" >$root_dir/index.html
chown -R apache. $root_dir

#根据需求修改配置文件
cat >> $conf<<END
NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot $root_dir
    ServerName $hostname
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
END

#启动服务并且开机自启动
service httpd start >/dev/null 2>&1
chkconfig httpd on

#测试验证（自己完成）
```

DNS

DNS 域名解析协议 Domain Name Systm

tcp 53

udp 53

正向 A 域名 ---->IP

反向 PTR IP----->域名

PS： MX表示邮件解析记录； *表示泛解析；@表示无主机名的直接解析。 解析条目中的IN可以省略不写。

原理

客户端本地缓存 -------->/etc/hosts ------>/etc/resole.conf ---->路由缓存----->本地DNS服务器---->转发DNS服务器--->根服务器

跟域名服务器 ---->顶级域名服务器---->主域名服务器---->保存结果到缓存---->返回结果给客户端

工作方式

递归查询 只查一次，不管得不得到结果

迭代查询 一直查下去，直到知道结果

一 装

yum install -y bind bind-chroot bind-utils

提供服务 bind服务以mount --bind的方式运行在/var/name/chroot中

二 配置

如果要使用安全的chroot方式，

systemctl start named-chroot

在配置时所有的文件和目录都要基于/var/named/chroot 目录为根目录。

比如要修改配置文件/etc/named.conf ，在chroot模式就要修改/var/named/chroot/etc/named.conf

/etc/logrotate.d/named ----日志轮转

/etc/named.conf ----主配置文件
/etc/named.rfc1912.zones ----zone配置文件

/usr/sbin/named-checkconf ----校验配置文件二进制命令
/usr/sbin/named-checkzone ----校验zone文件二进制命令

/var/named ----记录域名关系的文件目录
/var/named/named.localhost ----正向解析A记录文件
/var/named/named.loopback ----反向解析A记录文件
/var/named/slaves ----从服务记录目录 修改主服务器后如皋想slaves服务器立马同步，可以把这目录下文件删除掉

【搭建服务的第一步】
vim /etc/named.conf

listen-on port 53 { 127.0.0.1; any; }; 监听所有IP的53号端口

allow-query { localhost; any; }; 提供所有人DNS解析

【搭建服务的第二编辑zone 文件】
[root@rhel7 ~]# vim /etc/named.rfc1912.zones
zone "uplooking.com" IN { ----注册uplooking.com域
type master; ----服务类型（主服务器）
file "/var/named/uplooking.com.zone"; ----A记录文件存放的位置和名称
allow-update { none; }; ----是否允许其他主机更新A记录文件（none不允许）
};

zone "0.25.172.in-addr.arpa" IN { ----注册反向域172.25.0 网段
type master; ----服务类型
file "172.25.0.rev"; ----反向解析文件位置（/var/named/172.25.0.rev）
allow-update { none; }; ----不允许更新
};

搭建服务的第三部
3、填写A记录文件
辅助模板
[root@rhel7 ~]# cd /var/named
正想解析
[root@rhel7 named]# cp named.localhost uplooking.com.zone -a ----需要拷贝属性 -a zone的文件的属组要是named
反向解析
[root@rhel7 named]# cp named.loopback 172.25.0.rev -a

修改正向解析文件
++
vim /var/named/uplooking.com.zone
$TTL 1D -----生存周期
@ IN SOA @ rname.invalid. (
---注册域（uplooking.com） internet 授权 dns服务器主机名 管理员邮箱
0 ; serial ----序列号
用于主从服务同步
1D ; refresh ----刷新时间
1H ; retry ----失败重试
1W ; expire ----有效期

```
                                3H )    ; minimum                                    ----最小时间
        NS      @
        A       127.0.0.1                       ----ipv4解析
        AAAA    ::1                      ----ipv6


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++
vim /var/named/uplooking.com.zone
$TTL 1D
@       IN SOA  dns.uplooking.com. root.uplooking.com. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       NS      dns.uplooking.com.

www     A       172.25.0.10
dns      A       172.25.0.10
vip      A       172.25.0.11
 @       A       192.168.10.254

@       MX 5    dns.uplooking.com.
mail    A       172.25.0.10

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++
反向解析
vim /var/named/172.25.0.rev

$TTL 1D
@       IN SOA  dns.uplooking.com. root.uplooking.com. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       NS      dns.uplooking.com.
10      PTR     www.uplooking.com.
10      PTR     dns.uplooking.com.
11      PTR     vip.uplooking.com.

@       MX 5    dns.uplooking.com.
10      PTR      mail.uplooking.com.


启动服务

                named-checkconf                 检测住配置文件
                named-checkzone                 检测zone文件
                systemctl restart named          重启服务


测试
vim /etc/resolv.conf                    添加DNS服务器
```

sameserver 172.25.0.10

[root@rhel6 ~]# nslookup
> server ----查看服务端 DNS服务器的IP server 172.25.0.10
Default server: 172.25.0.10
Address: 172.25.0.10#53
> www.uplooking.com ----正向解析
Server: 172.25.0.10
Address: 172.25.0.10#53

> 172.25.0.11 -----反向解析
Server: 172.25.0.10
Address: 172.25.0.10#53
172.25.0.rev
11.0.25.172.in-addr.arpa name = vip.uplooking.com.

DNS主从

DNS 主从

规划：
主服务器：172.25.0.10
从服务器：172.25.0.11
测试机：172.25.254.250

1、修改主服务器：172.25.0.10

修改注册域的配置文件

正向
zone "uplooking.com" IN {

```
        type master;                                    ---主服务器
        file "/var/named/uplooking.com.zone";
        allow-transfer { 172.25.0.11; any；};           指定同步的从服务器
        allow-update { none; };
};
                                反向
zone "0.25.172.in-addr.arpa" IN {
        type master;
        file "172.25.0.rev";
        allow-transfer { 172.25.0.11; };
        allow-update { none; };
};
```

2、从服务器配置DNS服务　　可以直接打包拷贝主服务器的配置文件

```
[root@rhel6 ~]# yum -y install bind                     ----安装软件

listen-on port 53 { 127.0.0.1; any; };
allow-query     { localhost; any; };

vim /etc/named.rfc1912.zones

zone "uplooking.com" IN {
        type slave;                                     ----指定服务类型时从服务器
        file "/var/named/slaves/uplooking.com.zone";    ----指定A记录文件获取位置和名称
        masters { 172.25.0.10; };                       ----指定主服务的ip
        allow-update { 172.25.0.10; };                  ----允许主服务器进行A记录的更新
};

zone "0.25.172.in-addr.arpa" IN {
        type slave;
        file "/var/named/slaves/172.25.0.rev";
        masters { 172.25.0.10; };
        allow-update { 172.25.0.10; };
};
```

启动服务：

```
service named restart
```

查看监听端口：

```
[root@rhel6 ~]# netstat -tnpl |grep :53
```

测试：

```
172.25.254.250

vim /etc/resolv.conf

nameserver      172.25.0.11

[root@foundation0 cherrytree]# nslookup
```

在主服务器添加多条A记录，并修改主服务器的A记录序列号，使从服务器知道主服务器有记录更新

++

DNS 别名

www　　A　　172.25.0.10

bbb　　NAME　www

CNAME　----设置别名

连续的域名解析：

foundationX.example.com　　172.25.254.X

vim /var/named/uplooking.com.zone

$GENERATE 1-100 www$ IN A 172.25.254.$

wwwX.uplooking.com　　172.25.254.X

```
[root@rhel7 ~]# nslookup
> www10.uplooking.com
Server:         172.25.0.10
Address:        172.25.0.10#53

Name:   www10.uplooking.com
Address: 172.25.254.10
```

练习

2. 要求，
DNS服务器的IP为192.168.122.199/24
要完成两个域名， www.shangguan.com　www.lovelinux.com 正解和反解。
两个域名都解析到你另一个存在的虚拟机的IP，IP指定为192.168.122.200/24
（虚拟机的网络连接为default NAT）

DNS　192.168.122.199
配置两个zone文件即可

第一步创建zone文件
vim /etc/named

```
zone  "shangguan.com" IN  {
        type  master;
        file  "shangguan.com.db";
};
zone  "lovelinux.com" IN  {
        type  master;
        file  "lovelinux.com.db";
};

zone "122.168.192.in-addr.arpa" IN {
         type master;
         file "122.168.192.db";
```

};
第二部配置zone文件 创建数据库文件
ls -R /usr/share/doc/bind-9.9.4/sample/
rpm -ql |grep bind-9.9.4
cd /var/named
cp /usr/share/doc/bind-9.9.4/sample/var/named/my.internal.zone.db ./upl.net.db

第三部给数据库文件权限并且修改属组

[root@dns named]# chmod 640 shangguan.com.db lovelinux.com.db 122.168.192.db
[root@dns named]# chgrp named shangguan.com.db lovelinux.com.db 122.168.192.db

named-checkconf
named-checkzone shangguan.com shangguan.com.db

3. 自学思考：
dns 服务器在没有自身数据库查询的情况下，将请求转发(forward)给另外一个dns服务器。 理解条件转发(forward first)和直接转发(forward only)

准备两个dns服务器：
dns01 192.168.122.101
dns02 192.168.122.199

dns01 的解析请求发送到 dns02
dns02 :
listen-on port 53 { any; };
allow-query { any; };
recursion no;

dns01:
options {

listen-on port 53 { any; };
listen-on-v6 port 53 { none; };

.....
allow-query { any; };
forwarders { 192.168.122.199; };
forward only;

forward 这个选项仅在转发服务器列表非空的情况下有意义。值为first时,即其缺省值,将使服务器 fist 会一直找直到找到结果 only 先找自己的找不到就装发，转发找不到，就回复找不着。
首先请求转发服务器— 并且如果它不回答问题时,服务器再自行查找回答。如果指定only,服务器只请求转发服务器。
forwarders 指定用于转发的IP地址。缺省时空列表(不转发)。
转发也可以按域来配置,允许全局转发选项被多种方式覆盖。你可以设置某个特定的域使用不同的转发服务器,或者使用不同的forward only/first行为,或者全都不转发

4. 进阶思考：
DNS服务器的IP有两个 192.168.122.199/24 ； 192.168.10.199/24
针对www.linuxshare.com 域名提供解析：
当客户端从122网段访问dns时，解析到192.168.122.200
当客户端从10网段访问dns时， 解析到192.168.10.200

提示 view

排错技巧

容易犯的错误：

服务器的时间一定要准确；

db文件中A 记录 ， www A 192.168.122.101 ， 注意www要顶格写，前面不要有空格；

修改配置后，忘了重启服务。

出错之后，可以

named-checkconf -z /etc/named.conf

或者通过

systemctl -l status named

去查看错误日志。

Apache

apache --WEB服务器,http:80 https:443

apache （httpd）
httpd是apache开源软件组织（基金会）的众多开源软件中的一款。因为httpd这个网站服务器（web server）的应用之广（全世界市场占有量最大）之大，所以，很多时候我们直接用apache称呼该web server。
还要理解的是，http本身是作为一个协议的名称，而实现该服务器端的软件还有比如nginx , IIS(WINDOWS平台)，lighttpd等等。

apache(html/cgi) lighttpd nginx tomcat(jsp) weblogic(jsp) jboss(jsp)
apache+php（php）
+tomcat
lamp (linux+apache+mysql+php)
lnmp

html/asp/asp.net/php/jsp
+++
www.ccb.com --| html http://www.ccb.com
jsp https://www.ccb.com

Apache是著名的开源软件项目
Apache是著名的Web服务器软件
Apache名称的原型为A Patchy Server
Apache项目由Apache软件基金会（ASF）负责管理和开发

Apache服务器的特点
开放源代码
跨平台应用，可运行于Windows和大多数UNIX\Linux 系统
支持Perl、PHP、Python和Java等多种网页编程语言
采用模块化设计
运行非常稳定
具有相对较好的安全性

Apache服务器的两个版本分支
Apache服务器目前同时维护1.X和2.X两个版本分支
1.X
目前最高版本是1.3，系统运行稳定
缺乏一些较新的功能

2.X
具有新的功能特性
与1.X的配置存在较大差别
没有1.X运行稳定

应用web服务器之前的准备工作： 配置系统
- 静态ip
- selinux开启/关闭（实验环境要求关闭）
- firewalld（实验环境要求关闭）
- hostname（唯一标识）
- yum（用来安装软件）
- date （确保服务器时间准确，在真实环境中通常需要ntp时间同步）

```
[root@rhel7 ~]# rpm -ql httpd |more
/etc/httpd                            ----根目录
/etc/httpd/conf                       ----主配置目录
/etc/httpd/conf.d          ----扩展配置目录          /vhost.conf  创建虚拟主机
/etc/httpd/conf.modules.d  ----模块配置目录
/etc/httpd/conf/httpd.conf  ----主配置文件          该ServerNamer 0. 0.0.0：80
/var/log/httpd                        ----日志目录
/etc/httpd/logs                       ----日志目录
/etc/httpd/modules                    ----模块目录
/usr/sbin/apachectl                   ----二进制命令
/usr/sbin/httpd                       ----二进制命令
/var/www                              ----页面发布目录          /var/www/html/index.html
/var/www/cgi-bin                      ----cgi脚本
/var/www/html                         ----静态页面

curl -I http://www.baidu.com          ----查看网站WEB服务器的头部信息

vim /etc/httpd/conf/httpd.conf

ServerRoot "/etc/httpd"               ----根目录
#Listen 12.34.56.78:80
Listen 80                             ----监听端口
Include conf.modules.d/*.conf         ----扩展配置目录
User          apache                  ----后台用户
```

Group apache ----组
ServerAdmin root@localhost ----管理员邮箱

#ServerName www.example.com:80 ----域名访问
SserverName 0.0.0.0：80
DocumentRoot "/var/www/html" ----页面目录

<Directory "/var/www/html"> ----页面发布选项开始
Options Indexes FollowSymLinks ----支持索引页面、链接页面
AllowOverride None ----不允许读取.htaccess验证文件
Require all granted ----允许所有人访问页面（拒绝所有人访问require all deny）
</Directory> ----页面发布选项结束

添加一个索引页面
在DocumentRoot 路径下写index.htnll

定义别名
Alias

119 DocumentRoot "/test"
120 Alias "/test" "/etc/sysconfig"
131 <Directory "/etc">

访问控制

158 Require all granted ----允许所有人访问
159 Deny from 172.25.0.11 ----拒绝某一个主机

pache本地帮助文档
[root@web_server ~]# yum install -y httpd-manual
重启httpd后，
浏览器访问：
http://192.168.122.109/manual/

vhosts

用法 可以一个IP 解析两个不同的域名

搭建网站在虚拟主机vhosts.comf 上面搭建可以不修改配置文件

vhost的模板
rpm -ql httpd |grep vhosts
/usr/share/doc/httpd-2.4.6/httpd-vhosts.conf
vim /usr/share/doc/httpd-2.4.6/httpd-vhosts.conf
vim /etc/httpd/conf/httpd.comf

目录标签的权限设置
<Directory /www/docs/sikao> sikao----->是网站的目录
AllowOverride none
Require all granted
</Directory>

编辑vhosts文件
vim /etc/httpd/conf.d/vhosts.comf

ServerName 0.0.0.0:80 一定要写
<VirtualHost *:80> 网站的IP
ServerAdmin root.example.com 管理员的邮箱
DocumentRoot "/www/docs/sikao" 网站的目录
ServerName www.sikao.com 网站的名字
ServerAlias www.think.com 网站的别名
<Directory /www/docs/sikao>
allowoverride none 不允许覆盖
Require all granted 是所有人都可以访问 （在保持<Directory />的权限为Require all denied不变的情况下，为每个虚拟主机的家目录添加授权）
</Directory>
ErrorLog "/var/log/httpd/sikao.com-error_log" 报错日志
CustomLog "/var/log/httpd/sikao.com-access_log" common 成功访问的日志 --- common 报错的别名
</VirtualHost>

创建网站的目录
mkdir -p /www/docs/sikao

重启服务
systemctl restart httpd

安全网站跳接、

#NameVirtualHost *:80 RHEL6上要写
ServerNamer 0.0.0.0：80
<VirtualHost *:80>
ServerName www.upl01.com
DocumentRoot /htdocs/upl01
</VirtualHost>
<VirtualHost *:443>
ServerName www.upl01.com
DocumentRoot /htdocs/upl01
SSLEngine on
SSLCertificateFile /etc/httpd/ssl/web.crt
SSLCertificateKeyFile /etc/httpd/ssl/web.key
</VirtualHost>

思考：仅对第一个虚拟主机做了ｓｓｌ，可不可以对多个虚拟主机做ｓｓｌ

2.

NameVirtualHost *:80
<VirtualHost *:80>
ServerName www.upl01.com
Redirect 301 "/" "https://www.upl01.com/"
</VirtualHost>

443部分同上

作业

练习：

以下https均采用自签名证书：
1. 安装httpd, 创建网站首页内容为“This is my test page”， 支持https访问;
2. 配置httpd的运行用户和组为www;
3. 通过命令行工具从客户端访问httpd， 观察访问日志的增加；
4. 创建两个基于域名的虚拟主机，分别是www.upl01.com www.upl02.com，首页内容不同；

进阶：
1. 虚拟主机www.upl01.com 支持https协议访问；
2. 访问http://www.upl01.com 时， 自动跳转到https。

cgi 页面认证

vim /etc/httpd/conf/httpd.conf

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Require all granted                允许访问/var/www/cgi-bin的目录。
</Directory>
```

默认情况下，httpd已经支持cgi的目录，只需要将cgi的脚本放到指定的目录即可。

```
[root@web_server html]# cd  /var/www/cgi-bin/
[root@web_server cgi-bin]# vim time.sh
#!/bin/bash

echo "Content-type: text/html"
echo ""

/bin/cat <<EOF3
<html>
<head><title>System time</title></head>
<body>
<h2 align="center">The time of this system is :
EOF3

/bin/date "+%F %X"

/bin/cat <<EOF4
</h2>
</body>
</html>
EOF4
```

[root@web_server cgi-bin]# chmod +x time.sh
访问效果：http://192.168.122.109/cgi-bin/time.sh

1. 目录索引

在网站目录中没有index.html的时候，自动显示当前目录下的文件列表。

<Directory "/var/www/html">

Options Indexes FollowSymLinks Indexes是对目录支持索引的选项，FollowSymLinks 支持软链接。

</Directory>

[root@web_server conf.d]# mv welcome.conf welcome.conf.backup
[root@web_server conf.d]# systemctl restart httpd
在主配置文件中， 默认对/var/www/html支持 Indexes 。 但是在welcome.conf 中又设置Options -Indexes 取消对Indexes的支持。

如果是在虚拟主机里配置：如下
[root@web_server ~]# mkdir /www/shangguan/secret
[root@web_server ~]# touch /www/shangguan/secret/{a.txt,b.mp3,c.flv,d.unknow}

[root@web_server cgi-bin]# vim /etc/httpd/conf.d/vhost.conf
<VirtualHost *:80>
DocumentRoot /www/shangguan
ServerName www.shangguan.com
ErrorLog logs/www.shangguan.com-error_log
CustomLog logs/www.shangguan.com-access_log common
<Directory "/www/shangguan/secret/">
Options Indexes
</Directory>
</VirtualHost>

页面认证

使用.htaccess验证
vim /etc/httpd/conf/httpd.conf
<VirtualHost 192.168.10.100:8080>
ServerAdmin root.example.com
DocumentRoot "/data1"
ServerName www.abc.com
ServerAlias www.def.com
ErrorLog "/var/log/httpd/www.abc.com-error_log"
CustomLog "/var/log/httpd/www.abc.com-access_log" common
</VirtualHost>
<Directory "/data1">
AllowOverride All
Options none
</Directory>

将验证文件.htaccess放置在发布目录中
vim /data1/.htaccess
authtype basic ---验证方式
authname "please input your name & passwd" ---验证提示信息

authuserfile /etc/httpd/conf.d/userpasswd ----用户和密码存放文件位置和名称
require valid-user ---允许验证通过的用户来访问目录

htpasswd

-c ----创建密码文件 添加第2个用户时不需要-c的参数，c是创建新文件的意思。
-b ----非交互式创建用户和密码
[root@web_server conf.d]# htpasswd -cm /etc/httpd/.htpasswd user01
[root@web_server conf.d]# htpasswd -m /etc/httpd/.htpasswd user02

[root@rhel7 ~]# htpasswd -c /etc/httpd/conf.d/userpasswd harry ----创建密码和用户文件

重启服务：
systemctl restart httpd

[root@rhel7 ~]# cat /etc/httpd/conf.d/userpasswd
harry:$apr1$5gxNRAaM$MNc6Vy2A0gmihtakI./J00

CA证书

CA ----证书发布机构
----证书（公钥、私钥）
目录 /etc/pki 放密钥的目录
2. 使用自签名的crt （自己生成key，自己给自己签名颁发证书）

root@rhel7 ~]# cd /etc/pki/tls/certs/

[root@rhel7 certs]# make server.key ----生成私钥 需要输入密码

[root@rhel7 certs]# openssl rsa -in server.key -out server.key ----去掉密码 要输入之前生成密钥的密码

[root@rhel7 certs]# make server.csr ----生成公钥

Country Name (2 letter code) [XX]:CN ----国家
State or Province Name (full name) []:GD ----省份
Locality Name (eg, city) [Default City]:GZ ----城市
Organization Name (eg, company) [Default Company Ltd]:uplooking.com ----公司
Organizational Unit Name (eg, section) []:www.uplooking.com ---组织
Common Name (eg, your name or your server's hostname) []:root.uplooking.com ----主机名
Email Address []:root@uplooking.com ---邮箱

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:www.uplooking.com ----公司名称

安装安全模块

yum -y install mod_ssl

openssl x509 -in server.csr -req -signkey server.key -days 365 -out server.crt ----生成证书
编辑模块

[root@rhel7 certs]# vim /etc/httpd/conf.d/ssl.conf
100 SSLCertificateFile /etc/pki/tls/certs/server.crt

107 SSLCertificateKeyFile /etc/pki/tls/certs/server.key

方式 1 ： 不推荐，慢。在询问是否生存csr时选NO
[root@webserver ~]# yum install crypto-utils
[root@webserver ~]# genkey --days 365 www.upl.net

方式2 ：
[root@webserver ~]# openssl req -new -x509 -nodes -out web.crt -keyout web.key
需要填写的部分：
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:Guangdong
Locality Name (eg, city) [Default City]:Guangzhou
Organization Name (eg, company) [Default Company Ltd]:Shangguan
Organizational Unit Name (eg, section) []:tech
Common Name (eg, your name or your server's hostname) []:www.upl.net
Email Address []:123456@qq.com

[root@webserver ~]# ls web*
web.crt web.key

[root@webserver ~]# mkdir /etc/httpd/ssl

[root@webserver ~]# cp web.crt web.key /etc/httpd/ssl/

[root@webserver ~]# vim /etc/httpd/conf.d/ssl.conf

SSLCertificateFile /etc/httpd/ssl/web.crt

SSLCertificateKeyFile /etc/httpd/ssl/web.key

root@webserver ~]# systemctl restart httpd

在客户端浏览器中访问：
https://www.upl.net/

查看证书信息。
——

3. 使用权威的CA机构颁发的crt （当然这里是模拟CA）

192.168.122.109 webserver
192.168.122.101 ca机构
192.168.122.1 客户端

第1步： web服务器生成加密的key
[root@webserver ssl]# openssl genrsa -out server.key 2048

第2步： web服务器生成签名请求csr
[root@webserver ssl]# openssl req -new -key server.key -out server.csr
需要填写的部分：
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:Guangdong
Locality Name (eg, city) [Default City]:Guangzhou
Organization Name (eg, company) [Default Company Ltd]:Uplooking
Organizational Unit Name (eg, section) []:System
Common Name (eg, your name or your server's hostname) []:www.upl.net
Email Address []:23456@qq.com

第3步： 在CA服务器上建立用于签名的环境
[root@ca ~]# /etc/pki/tls/misc/CA -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to '/etc/pki/CA/private/./cakey.pem'
Enter PEM pass phrase: 需要设密码
Verifying - Enter PEM pass phrase: 重复密码

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]:US
State or Province Name (full name) []:JZ
Locality Name (eg, city) [Default City]:XY
Organization Name (eg, company) [Default Company Ltd]:google
Organizational Unit Name (eg, section) []:tech
Common Name (eg, your name or your server's hostname) []:www.google.com
Email Address []:654321@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: 不需要设密码
An optional company name []: 直接回车
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/CA/private/./cakey.pem: 输入cakey.pem的密码
Check that the request matches the signature
Signature ok

..... 中间输出略

Write out database with 1 new entries
Data Base Updated

第4步： 把webserver上的csr签名请求文件发送给CA到其签名目录
[root@webserver ssl]# scp server.csr 192.168.122.101:/etc/pki/CA/

第5步： CA对csr文件进行签名
[root@ca CA]# mv server.csr newreq.pem
[root@ca CA]# /etc/pki/tls/misc/CA -sign
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for /etc/pki/CA/private/cakey.pem: 输入cakey.pem的密码
Check that the request matches the signature
Signature ok
..... 中间输出略
Sign the certificate? [y/n]:y
..... 中间输出略
1 out of 1 certificate requests certified, commit? [y/n]y

第6步： 把新的签名好的证书回传给webserver
[root@ca CA]# scp newcert.pem 192.168.122.109:/root/ssl/

[root@webserver ssl]# mv newcert.pem server.crt
[root@webserver ssl]# ls

server.crt server.csr server.key

[root@webserver ssl]# mv server.key server.crt /etc/httpd/

第7步： 在ssl.conf中引用即可。

第8步： 如果需要浏览器信任该CA签名的证书，就需要把/etc/pki/CA/cacert.pem 导入到浏览器的证书机构中。

思考： 如何做到用户输入http://www.shangguan.com 会自动跳转到https://www.shangguan.com

apache报错

如果服务启动成功 访问时找不到服务

1.没有创建网站根目录
2.没有域名解析

script

作业1：
写一个自动搭建apache服务的脚本，要求如下：
1、用户输入web服务器的IP、域名以及数据根目录
2、如果用户不输入则一直提示输入，直到输入为止
3、当访问www.test.cc时可以访问到数据根目录里的首页文件“this is test page”
搭建基于域名的虚拟主机：
1> 关闭防火墙和selinux
2> 配置yum源（本地|内网）
3> 软件三步曲（查看软件是否安装|安装（确定是否成功安装）|查看软件列表）
4> 了解配置文件 man 5 xxx.conf
5> 根据需求通过修改配置文件来完成服务的搭建
6> 启动服务|开机自启动
7> 测试验证

```
#!/bin/bash
conf=/etc/httpd/conf/httpd.conf                                  路径长的先定义变量
fun_web(){
input=""                                                         输入的变量默认为空
output="$1"                                                      输出变量是用户输入的参数
while [ -z $input ]                                              -z  为空
  do
        read -p "$output" input                                      相当与“   read -p “请输入IP”  input  ”
  done
echo $input                                                      输出变量
}

#获取用户所输入的ip、hostanme、root_dir并赋值给变量

ip=$(fun_web "请输入你的IP地址(10.1.1.1)：")
hostname=$(fun_web "请输入你的主机名(www.test.cc)：")
root_dir=$(fun_web "请输入你的数据根目录(/var/www/html)：")

#修改hosts文件将域名和ip对应起来
cat >>/etc/hosts<<end
$ip $hostname
end

#判断数据根目录是否存在并创建首页文件及修改权限
[ ! -d $root_dir ] && mkdir -p $root_dir
echo "this is test page" >$root_dir/index.html
chown -R apache. $root_dir

#根据需求修改配置文件
cat >> $conf<<END
NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.co
    m
    DocumentRoot $root_dir
    ServerName $hostname
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
END

#启动服务并且开机自启动
service httpd start >/dev/null 2>&1
chkconfig httpd on

#测试验证（自己完成）
```

VSFTP

网络工具：

wget -r --递归下载目录

##

没有xinetd服务的时候 要把监听端口开启
xinetd 服务的配置文件等号后面加个空格 保险 大括号对齐
用户的家目录不能有写的权限

##
ftp文件共享： man 5 vsftpd
客户端：IE，firefox,Windows 客户端SFTP(putty)
命令行客户端：lftp(匿名用户)，ftp(本地用户登陆)

FTP指的是一种协议（文件传输协议），而vsftpd是Linux下一种ftp服务器软件。还有其他的一些ftp软件，比如tftp,proftpd,pure-ftpd等等。
FTP是一个跨平台的文件共享服务， 在windows , Linux, macOS都有服务器端和客户端软件的支持。

FTP的根目录不能有写的权限 否则会报错500
工作方式： 1 主动 21号端口建立传输链接 20 号端口传输数据
（命令端口，用于登录 执行命令等操作） （数据端口，上传下载）
2 被动 21号端口建立数据连接。 大于1024的端口传输数据
FTP传输数据采用主动模式还是被动模式是由FTP的客户端来决定。

一 装
yum -y install ftp lftp 工具
lftp IP 匿名访问
lpwd 看路径
mget day* 下载文件
mirror test 创建文件
ftp IP 本地用户访问
pwd

yum install -y vsftpd 安装服务

二1 配 vsftpd 默认配置即是一个匿名的只读共享。

VSFTPD 扩展

基于不同ip （一个提供内部访问 一个外部访问）
1、生成新的配置文件 复制原来的配置文件重命名
2、生成新的启动脚本
3、两个ip地址
4、分别启动两个vsftpd脚本

1、复制配置文件
cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd2
.conf
2、修改配置文件
vim /etc/vsftpd/vsftpd.conf
listen=YES YES 就是vsftpd是一个独立服务 NO 就是把它交给托管服务 33333#####
listen_address=172.25.0.10 #333####### 指定监听的IP #####
#listen_ipv6=YES #######

vim /etc/vsftpd/vsftpd2.conf
listen=YES ----打开ipv4监听
listen_address=192.168.10.100 ----ipv4监听地址
#listen_ipv6=YES ----注释监听的ipv6地址

3、生成新的启动脚本文件
cp /usr/lib/systemd/system/vsftpd.service /usr/lib/systemd/system/vsftpd2.service

vim /usr/lib/systemd/system/vsftpd2.service
7 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd2.conf ----将调用的配置文件，修改为vsftpd2.conf

4、配置两个ip地址
ifconfig eth0 172.25.0.10
ifconfig eth0:0 192.168.10.100

5、启动服务
systemctl restart vsftpd.service
systemctl restart vsftpd2.service

6、测试
查看服务端的监听ip
[root@rhel7 ~]# netstat -tnpl |grep :21
tcp 0 0 192.168.10.100:21 0.0.0.0:* LISTEN 25633/vsftpd
tcp 0 0 172.25.0.10:21 0.0.0.0:* LISTEN 25540/vsftpd

测试172.25.0.10主机ftp是否能登陆
测试192.168.10.100主机ftp是否能登陆

基于不同端口
vim /etc/vsftpd/vsftpd2.conf

listen_port=2121

systemctl restart vsftpd2.service

root@rhel7 ~]# netstat -tnpl |grep :21
tcp 0 0 192.168.10.100:2121 0.0.0.0:* LISTEN 25759/vsftpd
tcp 0 0 172.25.0.10:21 0.0.0.0:* LISTEN 25540/vsftpd

基于虚拟用户 为了服务器的安全 让访问服务器的用户通过虚拟用户访问
1、创建虚拟用户数据库 （用户名和密码）
vim /root/logins.txt
baidu
123
qq
456
wangyi
789
2、安装生成数据库文件的软件包
[root@rhel7 ~]# yum -y install libdb-utils
生成数据库文件
[root@rhel7 ~]# db_load -T -t hash -f /root/logins.txt /etc/vsftpd/login.db -T 指定数据库

-t 指定加密算法 -f 数据库文件

3、使用pam模块进行验证

默认情况下，vsftpd使用的是pam（可热插拔的用户认证系统）的方式进行用户认证，而目前Linux系统本地用户登录采用的也pam管理。

即vsftpd和login是同一套用户体系（/etc/passwd /etc/shadow）

vim /etc/pam.d/ftp

auth required /lib64/security/pam_userdb.so db=/etc/vsftpd/login 检验用户名 auth 认证 required 认证通过的 pam_userdb.so 用pam模块加密

account required /lib64/security/pam_userdb.so db=/etc/vsftpd/login 检验密码

[root@rhel7 ~]# ls /lib64/security/pam_userdb.so - ---确认有pam验证模块

/lib64/security/pam_userdb.so

4、创建本地的virtual用户

[root@rhel7 ~]# useradd -d /home/ftpsite -s /sbin/nologin virtual

[root@rhel7 ~]# ll /home

drwx------. 3 virtual virtual 74 Aug 12 02:42 ftpsite

5、添加测试文件

[root@rhel7 ~]# cp /etc/hosts /home/ftpsite/

[root@rhel7 ~]# ll /home/ftpsite/

total 4

-rw-r--r--. 1 root root 251 Aug 12 02:43 hosts

[root@rhel7 ~]# chown virtual.virtual /home/ftpsite/hosts

6、修改配置文件 grep -v '^#' /etc/vsftpd/vsftpd.conf |grep -v '^$' > /etc/vsftpd/vsftpd.conf

vim /etc/vsftpd/vsftpd.conf

anonymous_enable=NO

local_enable=YES

write_enable=YES

local_umask=022

anon_upload_enable=NO

anon_mkdir_write_enable=NO

dirmessage_enable=YES

xferlog_enable=YES 日志

connect_from_port_20=YES 数据传输

xferlog_std_format=YES

listen=YES

pam_service_name=/etc/pam.d/ftp ----虚拟用户pam模块验证文件

chroot_local_user=YES ----禁锢用户家目录

guest_enable=YES ----允许虚拟用户访问

guest_username=virtual ----虚拟用户

virtual_use_local_privs=YES ----允许虚拟用户调用本地用户权限

pasv_enable=YES ----开启被动模式

pasv_min_port=30000 ----被动模式的最小端口

pasv_max_port=30999 ----被动模式的最大端口

userlist_enable=YES

tcp_wrappers=YES

7、重启服务

systemctl restart vsftpd.service

chmod u-w /home/ftpsite ----将虚拟用户家目录的写权限剔除

watch -d -n 1 "netstat -tnpl |grep ftp" ----监控被动模式端口在30000-30999范围内
Every 1.0s: netstat -tnpl |grep ftp Sat Aug 12 02:59:04 2017

tcp 0 0 172.25.0.10:30120 0.0.0.0:* LISTEN 26931/vsftpd
8、测试：
测试机：172.25.254.250

ftp 172.25.0.10
baidu ----虚拟用户
123 ----密码

++

local_root -----本地用户指定家目录
anon_root -----指定匿名用户家目录
max_clients=100 -----最大链接客户端
max_per_ip=2 -----每个ip地址允许连接的客户端个数

客户端端口访问：

ftp IP 空格 端口号

lftp IP:端口号

vsftpd扩展： 限速

1、在主配置文件声明虚拟用户的扩展配置目录
vim /etc/vsftpd/vsftpd.conf
user_config_dir=/etc/vsftpd/ew

2、创建扩展目录
mkdir /etc/vsftpd/ew

3、创建虚拟用户配置文件 指定用户的速度
[root@rhel7 ~]# vim /etc/vsftpd/ew/baidu
local_root=/home/ftpsite/baidu ----指定家目录
local_max_rate=50000 ----限速50K

[root@rhel7 ~]# vim /etc/vsftpd/ew/wangyi
local_root=/home/ftpsite
local_max_rate=0 ----不限速

4、创建虚拟用户家目录
mkdir /home/ftpsite/{baidu}

5、创建测试文件 1T=1024G 1G=1024M 1M=1024kb
dd if=/dev/zero of=/home/ftpsite/baidu/file bs=1M count=10 dd if 从/dev/zero 文件 拷贝到of=哪里的文件 bs 块大小 为1M count 多少块 10 块 所以生产一个10M的空文件
dd if=/dev/zero of=/home/ftpsite/qq/file bs=1M count=20
dd if=/dev/zero of=/home/ftpsite/file bs=1M count=200

6、启动服务

systemctl restart vsftpd.service

7、测试
测试机：172.25.254.250

baidu　　---用户限速
[kiosk@foundation0 Desktop]$ ftp 172.25.0.10
Connected to 172.25.0.10 (172.25.0.10).
220 (vsFTPd 3.0.2)
Name (172.25.0.10:kiosk): baidu
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (172,25,0,10,118,244).
150 Here comes the directory listing.
-rw-r--r--　　1 0　　0　　10485760 Aug 12 08:20 file
-rw-r--r--　　1 0　　0　　19 Aug 12 08:19 file1
226 Directory send OK.
ftp> get file1
local: file1 remote: file1
227 Entering Passive Mode (172,25,0,10,120,157).
150 Opening BINARY mode data connection for file1 (19 bytes).　　指定的用户的家目录中的文件就限制速度了
226 Transfer complete.
19 bytes received in 5.5e-05 secs (345.45 Kbytes/sec)
ftp> get file
local: file remote: file
227 Entering Passive Mode (172,25,0,10,119,10).
150 Opening BINARY mode data connection for file (10485760 bytes).
226 Transfer complete.
10485760 bytes received in 210 secs (50.01 Kbytes/sec)

ftp 的设置信息

root@rhel7 uplooking]# rpm -ql vsftpd
/etc/logrotate.d/vsftpd ----日志轮询
/etc/pam.d/vsftpd ----
验证模块 虚拟用户
/etc/vsftpd ----配置
目录
/etc/vsftpd/ftpusers ----用户
限制文件
/etc/vsftpd/user_list ----用户
限制列表
/etc/vsftpd/vsftpd.conf ----主配置文
件
/usr/lib/systemd/system/vsftpd.service ----启动脚本 配
置两个服务时要修改
/usr/sbin/vsftpd ----二进
制命令
/var/ftp ----发布
共享的根目录
/var/ftp/pub ----
发布的共享目录

vim /etc/vsftpd/vsftpd.conf

anonymous_enable=YES ----是否允许
匿名用户
local_enable=YES ----
是否允许本地用户登陆
write_enable=YES ----
是否允许写入
local_umask=022 ----
本地用户上传文件的反掩码值 上传文件权限644
#anon_upload_enable=YES ----
匿名用户是否允许上传文件
#anon_mkdir_write_enable=YES ----匿名用户创
建、写权限
anon_other_write_enable=YES 匿名用戶 修改，删除，重命名文件的
权限
匿名不能下载自己上传的文件
anon_world_readable_only=YES （默认情况下，匿名用户只能够下载全世界（所有人）可读的文件，而
匿名用户上传上去的文件，生成的权限默认是600，所以不能下载）
anon_world_readable_only=NO 解决的方法：
或者
anon_umask=022 (默認是077)
dirmessage_enable=YES ----目录信息
xferlog_enable=YES ----
是否开启日志
connect_from_port_20=YES ----开启数据
传输端口20
#chown_uploads=YES ----是否开启
属性转换
#chown_username=whoever ---将匿名用户
转换成本地的对应用户

#xferlog_file=/var/log/xferlog ----指定日志的位置和
名称(默认位置)
#data_connection_timeout=120 ----数据传输超时链接
ftpd_banner=Welcome to blah FTP service ----欢迎标签
#chroot_local_user=YES ----是否开启将本地用
户禁锢家目录 全部用户都禁锢

#chroot_list_enable=YES ----是否开启禁锢家目录的用户列表 只禁锢列表中额
(default follows)
#chroot_list_file=/etc/vsftpd/chroot_list ----指定限制禁锢用户的列表
anon_root=/share/ftp -----添加匿名用戶訪問的目錄
local_root=/var/srv ----改變本地用戶的根目錄

默认本地用户登录到自己的家目录，可以进行上传下载的操作都可以。
但是， 当你把用户限制在自己家目录时，对本地用户而言，家目录/home/liubei即是它的ftp的根目录，而本地用户对自己的家目录是可写的，那么和ftp的默认安全策略（不允许用户对ftp的根目录可写）相冲突，导致ftp无法登录。

如何解决：
1. 让本地的ftp根目录不可写（但是让用户的家目录都不可写，不符合实际需求。）
2. 改变ftp的本地用户根目录到另外一个只读目录。
local_root=/var/srv
mkdir /var/srv
user_list既可以作为白名单，也可以做为黑名单，取决于主配置文件中的userlist_deny的选项。
[root@geust02 vsftpd]# pwd
/etc/vsftpd
[root@geust02 vsftpd]# ls
ftpusers user_list
默认情况下，主配置文件中，
userlist_deny=YES
那么，在usre_list文件中用户将不能访问ftp，即黑名单。

如果userlist_deny=NO
那么，在usre_list文件中用户将可以访问ftp，即白名单。

同时还检查用户名是否在ftpusers文件中，主要放了一系统服务的用户，这个文件中的用户永远都不能访问ftp，不管user_list是白还是黑。

练习

实例1：禁止匿名用户登陆
anonymous_enable=NO

实例2：禁止本地用登陆

local_enable=NO

实例3：允许匿名用户上传文件
anon_upload_enable=YES
anon_mkdir_write_enable=YES

实例4：禁锢用户家目录
chroot_local_user=YES ---禁锢所有用户 （默认情况下，本地用户可以登录FTP后，可以切换到别的系统目录去。这样很不安全。）

实例5：单独禁锢指定的本地用户
chroot_list_enable=YES ----开启禁锢列表
# (default follows)
chroot_list_file=/etc/vsftpd/chroot_list ----指定禁锢列表位置和名称

修改禁锢列表
vim /etc/vsftpd/chroot_list
cw03、

vsftpd应用实例5： FTP托管模式

如果vsftpd的服务不是一个频繁使用的服务，没有必要长期运行在系统中，占用系统资源，使用托管模式，有访问的时候才启动。
在Linux系统使用xinetd服务托管其他服务，而xinetd有很多的安全配置选项，使用得服务更安全。

对于xinetd服务

练习：
１．匿名用户可以上传，删除，下载，改名文件和目录。
２． 改变匿名用户登录的默认目录;
3. 实现将本地用户登录后限制在自己的家目录（在/home/username）
4. 改变本地用户登录的默认目录，所有用户登录后到同一目录（非家目录）。 （可选作业：如果不同的用户登录到不同的目录（非家目录），怎么做。提示：使用用户子配置文件。
5. 设置ftp独占模式下（非托管），下载客户端连接数量（最多支持2个客户端连接），传输速度的控制（下载速度限制在20K/S）。
6. 配置ftp使用xinetd的托管默认实现只在规定的时间（15：00－17：00）可以使用ftp的服务。
7. 了解vsftpd的虚拟用户用法和设置（不做要求）。

思考結論

扩展思考：如果非要对ftp的根目录上传文件，怎么做？

通过上面的配置和测试，发现vsftpd匿名用户要上传文件和目录有很多限制，需要一步步开放权限。
基本原则就是服务本身开放可写的同时，要开放上传文件夹的权限。

ftp 信息

500 就是ftp的目录有写的权限 服务会起不来 如果禁锢用户到家目录 那么 用户的家目录会变成根目录
550 是用户对文件夹没有操作的权限

881 bytes transferred 881字节传输成功

xinetd 托管服务

用xinetd来托管服务的好处是，可以利用xinetd守护进程的特性，访问控制，流量限制，日志增强，应用防火墙等等。还可以节省系统开销。对于访问量不大不频繁的ftp服务器，可以使用托管模式。

依赖服务 ： 依赖于xinetd服务运行的服务叫做依赖服务
独立服务：独立运行

如果想把某个服务变成依赖服务 就先把这个服务停掉 然后再/etc/xinetd.d/下面建该服务的的配置文件

/usr/lib/systemd/system/sshd.service ----独立服务的运行文件
/usr/lib/systemd/system/sshd.socket ----依赖服务的运行文件
需要的时候才启动

装
yum install -y xinetd

配 （相应的服务修改相应的配置文件） 停掉服务 交给xinetd 服务 托管启动
例子1
配置：
[root@geust02 ~]# systemctl stop vsftpd 关闭服务
[root@geust02 ~]# systemctl disable vsftpd 永久关闭

vim /etc/vsftpd/vsftpd.conf
listen=NO
listen_ipv6=NO

root@geust02 ~]# vim /etc/xinetd.d/vsftpd
service ftp
{
disable = no
socket_type = stream
wait = no
user = root
server = /usr/sbin/vsftpd
server_args = /etc/vsftpd/vsftpd.conf
}

启动
systemctl restart xinetd

例子2
ssh的托管

vim /etc/xinetd.d/sshd
service ssh
{
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/sbin/sshd
server_args = -i
log_on_failure += USERID
disable = no ----开启服务
only_from = 172.25.0.10 ----访问控制（仅允许）
no_access = 172.25.254.250 ----拒绝访问
}

service sshd stop ----将独立服务停止

service xinetd restart ----启动依赖服务

SSH

SSH 远程登陆
SSH 为 Secure Shell 的缩写，由 IETF 的网络工作小组（Network Working Group）所制定；SSH 为建立在应用层和传输层基础上的安全协议。SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH最初是UNIX系统上的一个程序，后来又迅速扩展到其他操作平台。SSH在正确使用时可弥补网络中的漏洞。SSH客户端适用于多种平台。几乎所有UNIX平台—包括HP-UX、Linux、AIX、Solaris、Digital UNIX、Irix，以及其他平台，都可运行SSH。
SSH在连接和传输数据的过程受用双向非对称加密。
sshd 已经默认开机启动。 sshd做为一个提供网络连接的服务，它需要监听网络接口和服务端的端口。
sshd 默认监听本机上所有的网络接口的ipv4 和 ipv6 的所有的地址， 默认监听22号端口，采用的tcp协议。（PS：每个服务默认规定使用的端口可以通过/etc/services文件查看）

```
[root@rhel7 ~]# yum list |grep openssh
openssh-clients.x86_64              6.4p1-8.el7             @anaconda/7.0         ----客户包
openssh-server.x86_64               6.4p1-8.el7             @anaconda/7.0         ----服务包

[root@rhel7 ~]# rpm -ql openssh-server
/etc/pam.d/sshd                                 ----验证文件
/etc/ssh/sshd_config                            ----主配置文件
/etc/sysconfig/sshd                             ----额外配置文件
/usr/lib/systemd/system/sshd.service      ----启动脚本
/usr/sbin/sshd                                  ----二进制命令
/usr/sbin/sshd-keygen                           ----生成密钥二进制命令

vim /etc/ssh/sshd_config
Port 22                         ----监听端口
```

实例1：修改监听端口 小于1024的被默认定义为服务端口）是最容易被攻击的对象，所以通常要改为一个大于1024的随机端口。）

```
Port 2222
systemctl restart sshd

[root@rhel7 ~]# netstat -tnpl |grep :22
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN      1470/sshd
tcp6       0      0 :::2222                 :::*                    LISTEN      1470/sshd
```

客户端访问：

```
[kiosk@foundation0 Desktop]$ ssh root@172.25.0.10 -p 2222
```

实例2：修改监听地址

```
AddressFamily inet                      从实际应用考虑，为安全性，只监听ipv4地址
ListenAddress 172.25.0.10               明确只监听某个IP接口（比如只监听内网的接口）

[root@rhel7 ~]# netstat -tnpl |grep :22
tcp        0      0 172.25.0.10:22          0.0.0.0:*               LISTEN      2613/sshd
```

LoginGraceTime 1m 减少最大的登录时长，但要合理，不能太小，太小的话正常请求都有可能登陆不

上。
PermitRootLogin no　因为root用户名固定容易被攻击，通常不允许root远程登录，而使用普通用户登录，普通用户名可以随机设置为较复杂（包括密码）。
#StrictModes yes
MaxAuthTries 6　重试次数
MaxSessions 10　最大会话数

AllowUsers zhangsan　只允许普通用户zhangsan远程登录。

UseDNS no　不要使用DNS解析主机名
……

其他配置选项，参见， man 5 sshd_config
对于sshd的安全加固注意事项：
1. 修改了ssh的端口前一定确保防火墙放行了新的端口号，否则应用生效后会导致连接不上。
2.禁止root登录前，一定义确保系统中有可登录的普通用户;
3.ssh的管理通常要配合安全脚本（DenyHosts,fail2ban）或防火墙作到防暴力破解;
4.经常检查系统安全日志，关注安全登录情况。
测试：
使用192.168.10.100访问时，发现没有监听端口

ssh日志类型：
SyslogFacility AUTHPRIV　　----/var/log/secure

实例4：密钥验证：
1、生成密钥对
[root@rhel7 ~]# ssh-keygen　　----生成密钥　一直回车 -q 静默模式 -f /root/.ssh/id_rsa -N "" 不显示生成密钥的过程 密码默认为空
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):　　----密钥存放位置
Enter passphrase (empty for no passphrase):　　---密钥密码
Enter same passphrase again:　　---确认密码
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
ab:6a:f6:6e:eb:eb:6f:3e:5d:2c:d6:35:21:27:76:dc root@rhel7
The key's randomart image is:
+--[RSA 2048]----+
|　　. . |
|　　+ = E|
|　　. = . |
|　　o |
|　S o . . |
|　.o + |
|　.o o |
|　o ..o . |
|　o.BO*o. |
+----------+
[root@rhel7 ~]# ls /root/.ssh/　　----查看密钥对
authorized_keys id_rsa id_rsa.pub known_hosts

2、拷贝公钥到被访问服务器
1）
[root@rhel7 ~]# scp /root/.ssh/id_rsa.pub root@172.25.0.11:/root/.ssh/authorized_keys
root@172.25.0.11's password:
id_rsa.pub　　100% 392 0.4KB/s 00:00

2）[root@foundation0 Desktop]# ssh-copy-id -i /root/.ssh/id_rsa.pub root@172.25.0.11　只打红色部分的就可以把密钥发送过去了
PS，ssh-copy-id不支持非22号端口，如果已改变ssh端口，可临时改回默认，上传完公钥后再改成特殊端口，不会产生影响。又或者使用scp或rsync的文件上传功能替换ssh-copy-id。
ssh 连接非22端口服务器的方法：

ssh -p 29966 root@远程ssh服务器的ip #小写p
scp 远程拷贝非22端口的服务器文件的方法:
上传文件到服务器
scp -P 29966 /Users/ianMac/Desktop/progit.zh.pdf root@远程ssh服务器的ip:/home/wwwroot #大写P

3、测试：

rhel7--172.25.0.10　　---访问172.25.0.11 免密码登陆

利用pam模块，限制用户sshd登陆失败次数：

vim /etc/pam.d/sshd
　　auth　required pam_tally2.so　deny=3　　unlock_time=300　　ven_deny_root root_unlock_time=200

　　deny=3　　---失败次数
　　unlock_time=300　　---锁定时间
　　even_deny_root　　---root用户登陆失败，也一并拒绝
　　root_unlock_time　　----root锁定时间
测试：
在172.25.0.10主机查看：
　　pam_tally2 -u　　student　　----查看
　　pam_tally2 -r　-u　　student　　----清空用户登陆失败次数
[root@rhel7 ~]# pam_tally2 -u student
Login　　Failures Latest failure　　From
student　　3　　01/04/02 00:05:13　rhel6-f0.example.com

safety consolidaton

1.修改监听端口
　　/Port　4600
2。禁止root用户登陆
　　/Root ----→ no
3. 修改监听地址
　/ListenAddres　本机的网卡地址　如192.168.3.1
4. 禁止DNS解析　远程链接的时候会快一点
　　/UseDNS　---→　　no
5.只有你想远程链接主机 上的用户如（user01）的家目录（/home/user01/.ssh）下有authorized_keys　这个文件 那么 你客户端上的任意用户都能免密码远程登陆

error

Warning: Permanently added '192.168.3.3' (ECDSA) to the list of known hosts.
Connection closed by 192.168.3.3
链接拒绝
　　删掉要远程的机子的/root/.ssh/known_hosts　文件

aotufs

触发挂载

装
　　yum -y install autofs

　配
修改自动挂载配置文件

/etc/auto.master　　----主配置文件
/etc/auto.misc　　----子配置文件

vim /etc/auto.master
　　/misc　　/etc/auto.misc
　　挂载目录　　子配置文件名称

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

linux -ro,soft,intr ftp.example.org:/pub/linux

例子
1)修改主配置文件
vim /etc/auto.master
/data /etc/auto.data
挂载目录 子配置文件
/data /etc/auto.data --timeout=5 ----(超时连接)
2）生成子配置文件
cp /etc/auto.misc /etc/auto.data
3)修改子配置文件
vim /etc/auto.data

uplooking -fstype=xfs,rw :/dev/sdb3 （触发挂在磁盘
触发点 挂载选项 挂载磁盘
uplooking -ro,soft,intr 172.25.254.250:/var/ftp/pub （触发挂载服务
4)启动服务
systemctl restart autofs
5）测试
mkdir /data/uplooking -p
cd /data/uplooking ----触发触发点
df -h
/dev/sdb3 2.0G 33M 2.0G 2% /data/uplooking

虚拟内存
1、生成新的存储空间
fdisk /dev/sdc
n
p
1

+1G
t
1
82
w

刷新
partx -a /dev/sdc
格式化
mkswap /dev/sdc1
临时挂载：
swapon /dev/sdc1
swapon -s ---查看

永久挂载swap虚拟内存
vim /etc/fstab
/dev/sdc1 swap swap defaults 0 0

例子

useradd -d /nfs/user01 -M user01
useradd -d /nfs/user02 -M user02
useradd -d /nfs/user03 -M user03

echo 123 |passwd -stdin user01
echo 123 |passwd -stdin user02
echo 123 |passwd -stdin user03

vim /etc/auto.master
/nfs /etc/auto.nfs
自动挂载的服务
cp /etc/auto.misc /etc/auto.nfs

vim /etc/auto.nfs
user01 -rw,soft,intr 172.25.0.10:/nfs/share1
user02 -rw,soft,intr 172.25.0.10:/nfs/share2
user03 -rw,soft,intr 172.25.0.10:/nfs/share3
觸發點 選項

service autofs restart
测试：

使用测试机f0访问服务器2的user01，user02，user03用户触发rhel7的服务器1共享目录的挂载
[user01@rhel6 ~]$ ll
total 0
-rw-rw-r--. 1 user01 user01 0 Aug 8 13:50 abc
[user01@rhel6 ~]$ df -h
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/vg_rhel6-LogVol01 17G 3.0G 14G 19% /
tmpfs 499M 224K 499M 1% /dev/shm
/dev/vda1 485M 39M 421M 9% /boot
/dev/mapper/vg_rhel6-LogVol00 388M 11M 358M 3% /home
172.25.0.10:/nfs/share1 18G 3.0G 15G 18% /nfs/user01

iptables&squid

iptables
Linux的软件防火墙工具由Netfilter的内核模块提供功能。
iptables 用户空间防火墙管理工具。
在RHEL6使用的是iptables
在RHEL7上面使用iptables 或者 firewalld

3表5链

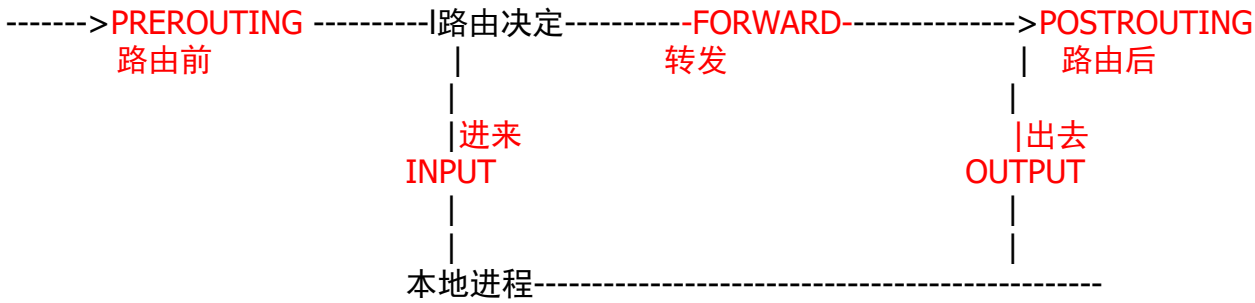
	表		
链	fitter	nat	mangle
INPUT	有		有
OUTPUT	有		有

FORWARD 有 有 有

POSTROUTING 有 有

PREROUTING 有 有

工作流程



按顺序排列规则，按从上到下的顺序匹配，匹配后立即退出
规则可以指定多个匹配条件
必须满足规则说明中的每个条件才算是匹配 如果无匹配规则则应用链的默认策略
如果在定制链中找不到匹配，那么就会返回到调用这个定制链的链

表的优先级
raw ------->mangle--------->nat---------->filter

filter 默认表。数据包过滤
nat 数据包的转发 地址转换
mangle 数据包修改

链
INPUT 本机进站的数据流
OUTPUT 本机出站的数据流
FORWARY 路由的数据流
POSTROUTING 路由后的数据流
PREROUTING 路有前的数据流

保存路由规则
iptables-save >/tmp/iptables.save ----保存到指定路径
iptables-restore < /tmp/iptables.save ----根据文件恢复策略

iptbles -t filter（表） -A（选项） -m multiport （多个端口 -p（协议） -m -d（IP） -dport （目的端口） -j （动作）
nat -s -sport （原端口）
选项
-A ----添加规则
-D ----删除一条规则
-C ----修改
-I ----插入规则
-L ----list
-F ----清空防火墙策略
-P ----设置默认规则 iptables -t filter -P INPUT DROP 把INPUT链的默认动作改成丢弃
-R ----替换规则 iptables -t filter -R INPUT 1 -m mac --mac-source 52:54:00:00:00:0B -p icmp -j ACCEPT
-n -----不解析
-i ---in 进站网卡

-o ----out 出站网卡 iptables -t nat -A POSTROUTING -o eno33554984 -s 172.25.254.0/24 -j MASQUERADE

iptables -t nat -A POSTROUTING -o eno33554984 -j SNAT --to-source 10.10.10.10

-p ----指定协议 icmp tcp udp
--dport ----指定目的端口
--sport -----指定源端口
-s ----源ip地址
-d ----目的ip地址
-src ----源
-dst ---目的

-j ---指定动作
ACCPET ---接收
REJECT ---拒绝
DROP ---丢弃

连续的如（20 21 23 24） 20：25

-m multiport 指定多个端口 不连续的如（21 80）21，88
-m mac --mac-source 52:54:00:00:00:0B 指定MAC地址
-m iprange --src-range 172.25.0.100-172.25.0.251 指定多个IP地址
-m state --state INVALID iptables -A INPUT -m state --state INVALID -j DROP 丢弃所有到达防火墙的无效的访问

iptables -A OUTPUT -d 172.25.0.10 -p tcp --dport 21 -j REJECT

vim /tmp/123

*filter ----添加表
-A INPUT -p tcp -m iprange --src-range 172.25.254.10-172.25.254.20 -m multiport --dport 80,20,21,110,995,143,993 -j ACCEPT

-A INPUT -p tcp -m mac --mac-source 52:54:00:00:00:0B --dport 22 -j ACCEPT
-A INPUT -s 172.25.0.0/24 -p tcp --dport 25 -j ACCEPT
-A INPUT -p tcp -s 172.25.0.0/24 --dport 53 -j ACCEPT
-A INPUT -p udp -s 172.25.0.0/24 --dport 53 -j ACCEPT
-R INPUT 3 -p tcp -s 172.25.0.11 --dport 25 -j REJECT
-P INPUT DROP
COMMIT ----添加结束

iptables-restore < /tmp/123 ----恢复规则

1、开启路由转发功能
临时开启
echo 1 >/proc/sys/net/ipv4/ip_forward
永久开启
vim /etc/sysctl.conf
net.ipv4.ip_forward = 1
sysctl -p 立即生效

SNAT---源地址转换
内网 ----> SNAT 服务器 ---> 外网
内部网络 外网
DNAT---目的地址转换
客户端---->baidu页面 服务器DNAT ----> 页面服务器
外网 内部网络

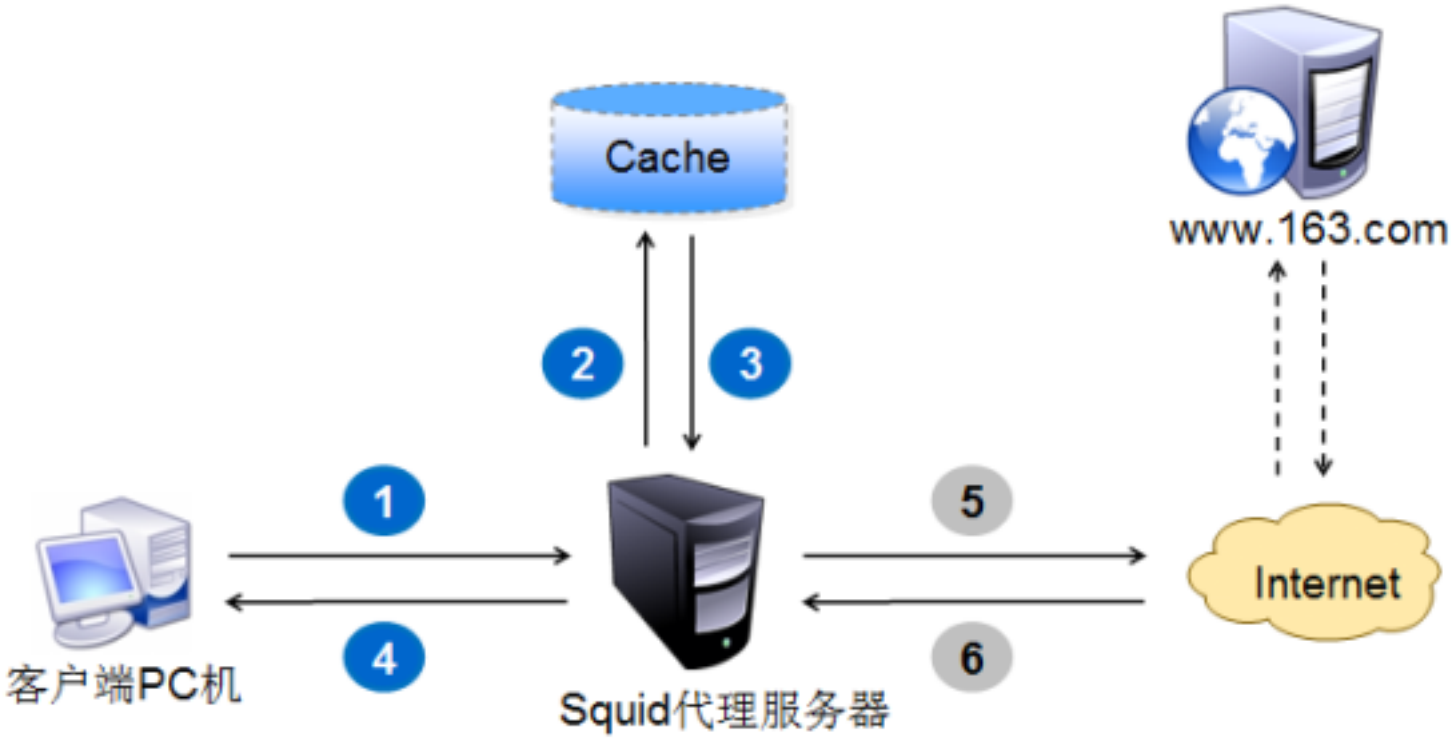
squid --代理服务器,支持http/ftp 3128/tcp

1.WEB加速 缓存静态页面

2.访问控制 添加acl

3.提高安全性,隐藏内网 内网通过代理上网 正向代理 反向代理

4.翻墙(GWF)



TCPwrappers

（防火墙）　　——过滤TCP包头(/usr/sbin/tcpd)

匹配顺序: tcp-->tcpwrappers-->hosts.allow--> hosts.deny,默认情况下这两个文件是空的,规则马上写马上生效.

1、如果在hosts.allow能够匹配到相应的规则，则允许，匹配到此结束。

2、如果在hosts.allow匹配不到相应规则，接下来匹配hosts.deny文件，如果匹配到则拒绝，匹配到此结束。

3、如果在hosts.allow和hosts.deny中都无法匹配到相应规则，则允许。

vim /etc/hosts.allow　　----允许

vim /etc/hosts.deny　　----禁止

实例1：

vim /etc/hosts.allow

sshd:　　ALL　　EXCEPT:172.25.0.11

放通所有人，但是拒绝172.25.0.11的主机访问

实例2：

将访问用户进行记录：

vim /etc/hosts.allow

sshd:　　172.25.0.11:spawn echo "login attempt from %u@%h to %s %p" |mail -s "info about sshd login" root@localhost

spawn　　:操作动作

login attempt from %u@%h to %s %p　　：发送邮件的正文

%c　----client

%u　----user

%s　----server

%p　----pid

info about sshd login　　----邮件主题

root@localhost　　----接收人邮箱

看邮件

tailf /var/mail/root

用的上规则

iptables -A INPUT -m state --state INVALID -j DROP

丢弃所有到达防火墙的无效的访问；

iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

放行所有到达防火墙的由其他已经通过防火墙的连接数据包产生的关联包或者后续保持连接的包；

iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT

访问80端口，但是只放行第一个数据包，后续的数据包在前一条规则已经可以通过，以提高效率；

iptables -A INPUT -i lo -j ACCEPT
放行内部连接必须环回接口，即127.0.0.1

iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --name ssh --update --seconds 600 --hitcount 5 -j DROP
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --name ssh --set
上面两条规则，对22端口的连接进行安全加固，在10分钟之内，只能发起5次新的连接请求，超过会被丢弃。

iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
放行正常的ssh 22端口的请求。
i
ptables -A INPUT -j REJECT --reject-with icmp-host-prohibited
拒绝所有其他进入防火墙INPUT链的连接，拒绝信息是主机禁止访问。

iptables -A FORWARD -j REJECT --reject-with icmp-host-prohibited
拒绝所有其他进入防火墙FORWARD链的连接，拒绝信息是主机禁止访问。

对于OUTPUT链没有规则，默认放行。

-m state --state NEW 可以用--syn 来代替。

在RHEL7 如何持久化保持iptables 的规则：

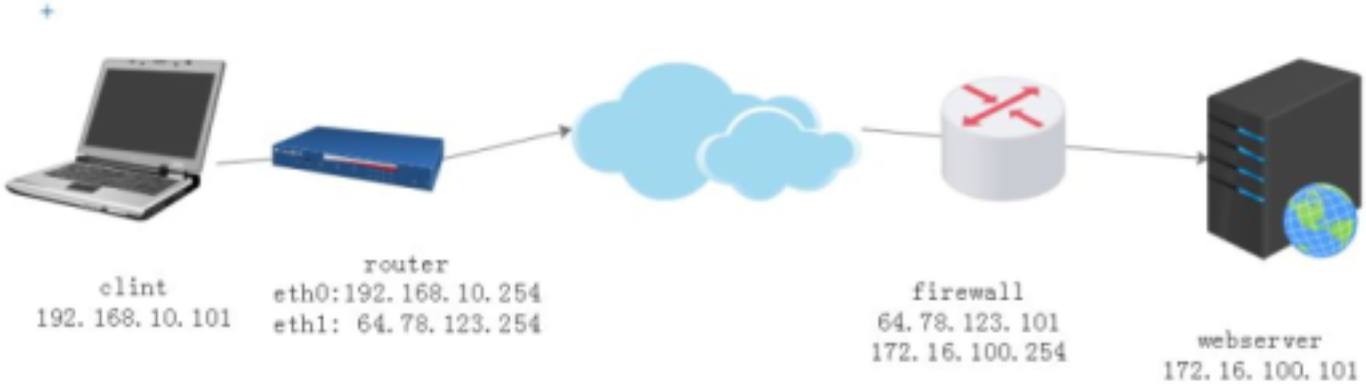
```
[root@geust02 ~]# systemctl stop firewalld
[root@geust02 ~]# systemctl disable firewalld
[root@geust02 ~]# yum install -y iptables-services
[root@geust02 ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@geust02 ~]# systemctl enable iptables
```

重启后规则仍然生效。

iptables -t filter -A INPUT -p tcp -s 172.25.0.0/24 --dport 20:25 -j ACCEPT ----连续端口

iptables -A INPUT -m multiport -p tcp --dports 22,80,110 -j ACCEPT ----多个端口

练 习



1.配置WEB服务器的防火墙。
要求：在主机上安装httpd ,vsftpd软件，并启动相应的服务，配置以下防火墙策略，使服务能被客户端正常访问。
1.拒绝所有的无效包的连接请求。(filter INPUT)
2.对于80,21的端口的请求放行。(filter INPUT)
3.对于22端口的请求只允许eth0上的IP联接，假设还有eth1.(filter INPUT)
4.对于22端口的非法连接请求记录日志。(filter INPUT)
5.禁止服务器被ping。(filter INPUT)
6.禁止服务器主动向外发送联网请求，但可以进行ping外网测试。(filter OUTPUT)
7.拒绝其他的所有链接。(filter policy)

2.配置DNAT/SNAT的转发防火墙。
环境：准备两个虚拟机，vhost01 vhost02. 注意要停掉真机本身的防火墙（清空）。
vhost01 有两张网卡，eth0 桥接到br0，IP: 192.168.10.x eth1 采用默认的NAT连接 IP:192.168.122.x 。
vhost02 有一张网卡，eth0 采用默认的NAT连接 IP: 192.168.122.y。在vhost02上安装httpd服务器，实现访问。
要求：
1.找一个网段在192.168.10.0/24的机器(非虚拟机所在的机器)做测试客户端。
2.使用vhost01做DNAT防火墙，将请求到vhost01的80端口的请求转发到vhost02上去。(nat PREROUTING)(filter FORWARD)
3.将请求vhost01 4567端口的请求转发到vhost02 22号端口上去。(nat PREROUTING)(filter FORWARD)
4.vhost02能够主动ping外网（前提vhost01可以上外网）。(nat POSTROUTING)(filter FORWARD)

补充：

关于虚拟机网卡的桥接：

默认NAT

配桥接：
方法1（操作简单）：
[root@huangdaojin ~]# /etc/init.d/NetworkManager stop
停止 NetworkManager 守护进程：   [确定]
[root@huangdaojin ~]# chkconfig NetworkManager off

[root@huangdaojin ~]# virsh iface-bridge eth0 br0

[root@huangdaojin ~]# /etc/init.d/network restart

结果：

```
[root@huangdaojin ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
HWADDR=00:16:41:E1:5B:76
ONBOOT="yes"
BRIDGE=br0
[root@huangdaojin ~]# cat /etc/sysconfig/network-scripts/ifcfg-br0
DEVICE=br0
ONBOOT=yes
TYPE=Bridge
BOOTPROTO=none
IPADDR=192.168.10.58
NETMASK=255.255.255.0
GATEWAY=192.168.10.1
STP=on
DELAY=0

[root@huangdaojin ~]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 223.5.5.5
nameserver 223.6.6.6


[root@huangdaojin ~]# ip addr show

9: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 00:16:41:e1:5b:76 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.58/24 brd 192.168.10.255 scope global br0
    inet6 fe80::216:41ff:fee1:5b76/64 scope link
       valid_lft forever preferred_lft forever
```

——

方法2：

```
[root@365linux network-scripts]# cp ifcfg-eth0 ifcfg-br0
[root@365linux network-scripts]# ifdown eth0

[root@365linux network-scripts]# vim ifcfg-eth0
DEVICE="eth0"
BOOTPROTO=none
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
UUID="797b705d-af54-4bac-b726-89be08589e27"
HWADDR=00:16:41:E1:43:0D
BRIDGE=br0

[root@365linux network-scripts]# vim ifcfg-br0
DEVICE="br0"
BOOTPROTO=none
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Bridge"
IPADDR=192.168.10.58
PREFIX=24
GATEWAY=192.168.10.1
DNS1=8.8.8.8
DNS2=8.8.4.4
```

```
[root@365linux network-scripts]# service network restart

[root@365linux ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:16:41:e1:43:0d brd ff:ff:ff:ff:ff:ff
    inet6 fe80::216:41ff:fee1:430d/64 scope link
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:16:cf:b9:8b:33 brd ff:ff:ff:ff:ff:ff
6: virbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 52:54:00:6c:45:32 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
7: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 500
    link/ether 52:54:00:6c:45:32 brd ff:ff:ff:ff:ff:ff
11: vnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen
500
    link/ether fe:54:00:9e:fb:7b brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fc54:ff:fe9e:fb7b/64 scope link
       valid_lft forever preferred_lft forever
12: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 00:16:41:e1:43:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.58/24 brd 192.168.10.255 scope global br0
    inet6 fe80::216:41ff:fee1:430d/64 scope link
       valid_lft forever preferred_lft forever
```

——

选择一个虚拟机添加（或者修改）网卡的源设备为eth0(桥接到br0).

```
[root@365linux ~]# brctl show
bridge name     bridge id               STP enabled     interfaces
br0             8000.001641e1430d       no              eth0
                                                        vnet2
virbr0          8000.5254006c4532       yes             virbr0-nic
                                                        vnet0
                                                        vnet1
```

1.配置WEB服务器的防火墙。
要求：在主机上安装httpd ,vsftpd软件，并启动相应的服务，配置以下防火墙策略，使服务能被客户端正常访问。
1.拒绝所有的无效包的连接请求。(filter INPUT)
2.对于80,21的端口的请求放行。(filter INPUT)
3.对于22端口的请求只允许eth0上的IP联接，假设还有eth1.(filter INPUT)
4.对于22端口的非法连接请求记录日志。(filter INPUT)
5.禁止服务器被ping。(filter INPUT)
6.禁止服务器主动向外发送联网请求，但可以进行ping外网测试。(filter OUTPUT)
7.拒绝其他的所有链接。(filter policy)

```
[root@vhost ~]# yum install httpd vsftpd
[root@vhost ~]# iptables -F
[root@vhost ~]# iptables -t nat -F
[root@vhost ~]# iptables -I INPUT -m state --state INVALID -j DROP
[root@vhost ~]# iptables -A INPUT -m state --state NEW -p tcp -m multiport --dports 80,21 -j ACCEPT
```

```
[root@vhost ~]# iptables -I INPUT 2 -m state --state RELATED,ESTABLISHED -j ACCEPT
[root@vhost ~]# iptables -I INPUT 3 -i lo -j ACCEPT
[root@vhost ~]# modprobe nf_conntrack_ftp
[root@vhost ~]# modprobe nf_nat_ftp
[root@vhost ~]# vim /etc/sysconfig/iptables-config
IPTABLES_MODULES="nf_conntrack_ftp nf_nat_ftp"
[root@vhost ~]# iptables -A INPUT -m state --state NEW -i eth0 -d 192.168.1.74 -p tcp --dport 22 -j
ACCEPT
[root@vhost ~]# iptables -A INPUT -p tcp --dport 22 -j LOG --log-tcp-options  --log-ip-options
[root@vhost ~]# iptables -A INPUT -p icmp --icmp-type 8 -j DROP  //可选
[root@vhost ~]# iptables -A INPUT -j REJECT --reject-with icmp-host-unreachable
[root@vhost ~]# iptables -A FORWARD -j REJECT --reject-with icmp-host-unreachable
[root@vhost ~]# iptables -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
[root@vhost ~]# iptables -I OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
[root@vhost ~]# iptables -A OUTPUT -j REJECT --reject-with icmp-host-unreachable
[root@vhost ~]# iptables -I OUTPUT 2 -o lo -j ACCEPT


[root@vhost ~]# iptables-save > /etc/sysconfig/iptables-2013-03-30
[root@vhost ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]

[root@vhost ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  0.0.0.0/0            0.0.0.0/0           state INVALID
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW multiport dports 80,21
ACCEPT     tcp  --  0.0.0.0/0            192.168.1.74        state NEW tcp dpt:22
LOG        tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22 LOG flags 6 level 4
DROP       icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 8
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 8
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-unreachable
```

测试：

```
[root@vhost ~]# service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for vhost
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for
ServerName
                                                           [  OK  ]
[root@vhost ~]# chkconfig httpd on
[root@vhost ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@vhost ~]# chkconfig vsftpd on

[root@teacher ~]# elinks http://192.168.1.74
```

```
[root@teacher ~]# ftp 192.168.1.74
Connected to 192.168.1.74 (192.168.1.74).
220 (vsFTPd 2.2.2)
Name (192.168.1.74:chuyue): ftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

[root@vhost ~]# ip addr add 192.168.1.75/24 dev eth0
[root@teacher ~]# ssh 192.168.1.75
ssh: connect to host 192.168.1.75 port 22: No route to host

[root@vhost ~]# tail /var/log/messages
Mar 30 12:37:21 localhost kernel: IN=eth0 OUT= MAC=52:54:00:8e:e2:82:50:46:5d:6f:ba:c4:08:00
SRC=192.168.1.100 DST=192.168.1.75 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=11945 DF
PROTO=TCP SPT=35621 DPT=22 WINDOW=14600 RES=0x00 SYN URGP=0 OPT
(020405B40402080A0049E3C90000000001030307)

[root@teacher ~]# ping 192.168.1.74

[root@vhost ~]# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.111 ms

[root@vhost ~]# nslookup www.baidu.com
```

2.配置DNAT/SNAT的转发防火墙。
环境：准备两个虚拟机，vhost01 vhost02. 注意要停到真机本身的防火墙。
vhost01 有两张网卡，eth0 桥接到br0，IP: 192.168.1.x eth1 采用默认的NAT连接 IP:192.168.122.x 。
vhost02 有一张网卡，eth0 采用默认的NAT连接 IP: 192.168.122.y。在vhost02上安装vsftpd服务器，实现匿名访问。
要求：
1.找一个网段在192.168.1.0/24的机器做测试客户端。
2.使用vhost01做DNAT防火墙，将请求到vhost01的21端口的请求转发到vhost02上去。(nat PREROUTING)(filter FORWARD)
3.将请求vhost01 4567端口的请求转发到vhost02 22号端口上去。(nat PREROUTING)(filter FORWARD)
4.vhost02能够主动ping外网（前提vhost01可以上外网）。(nat POSTROUTING)(filter FORWARD)

firewall : vhost01 eth0: 192.168.1.78 eth1 192.168.122.204
real server : vhost02 eth0: 192.168.122.71

```
[root@vhost01 ~]# iptables -t nat -I PREROUTING -d 192.168.1.78 -p tcp --dport 21 -j DNAT --to
192.168.122.71
[root@vhost01 ~]# iptables -I FORWARD -d 192.168.122.71 -p tcp --dport 21 -j ACCEPT
[root@vhost01 ~]# iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
[root@vhost01 ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
[root@vhost01 ~]# vim /etc/sysctl.conf
net.ipv4.ip_forward = 1

[root@vhost01 ~]# modprobe nf_conntrack_ftp
[root@vhost01 ~]# modprobe nf_nat_ftp

[root@vhost02 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
```

```
BOOTPROTO="none"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR=192.168.122.71
NETMASK=255.255.255.0
GATEWAY=192.168.122.204
[root@vhost02 ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
0.0.0.0         192.168.122.204 0.0.0.0         UG    0      0        0 eth0

[root@vhost02 ~]# service iptables stop
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
[root@vhost02 ~]# chkconfig iptables off

[root@vhost01 ~]# iptables -t nat -A PREROUTING -d 192.168.1.78 -p tcp --dport 4567 -j DNAT --to
192.168.122.71:22
[root@vhost01 ~]# iptables -I FORWARD 3 -d 192.168.122.71 -p tcp --dport 22 -j ACCEPT

4.
[root@vhost01 ~]# iptables -t nat -A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p tcp -
j SNAT --to-source 192.168.1.78:1024-65535
[root@vhost01 ~]# iptables -t nat -A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p udp -
j SNAT --to-source 192.168.1.78:1024-65535
[root@vhost01 ~]# iptables -t nat -A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -j
SNAT --to-source 192.168.1.78
[root@vhost01 ~]# iptables -I FORWARD 4 -s 192.168.122.0/24 ! -d 192.168.122.0/24 -j ACCEPT
```

SNAT&正向代理

内网 访问公网
SNAT： 源地址转发
172.25.254.240
172.25.254.241
172.25.254.242

172.25.254.241 ---------> 172.25.254.240 进站网卡eno16777736 SNAT服务器 出站网卡
eno33554984 10.10.10.10 -------> 10.10.10.1
内网主机 内网主机 公网主机 www.uplooking.com页面

在SNAT服务器
eno16777736:172.25.254.240
eno33554984:10.10.10.10
1、开启路由转发功能
临时开启
echo 1 >/proc/sys/net/ipv4/ip_forward
永久开启

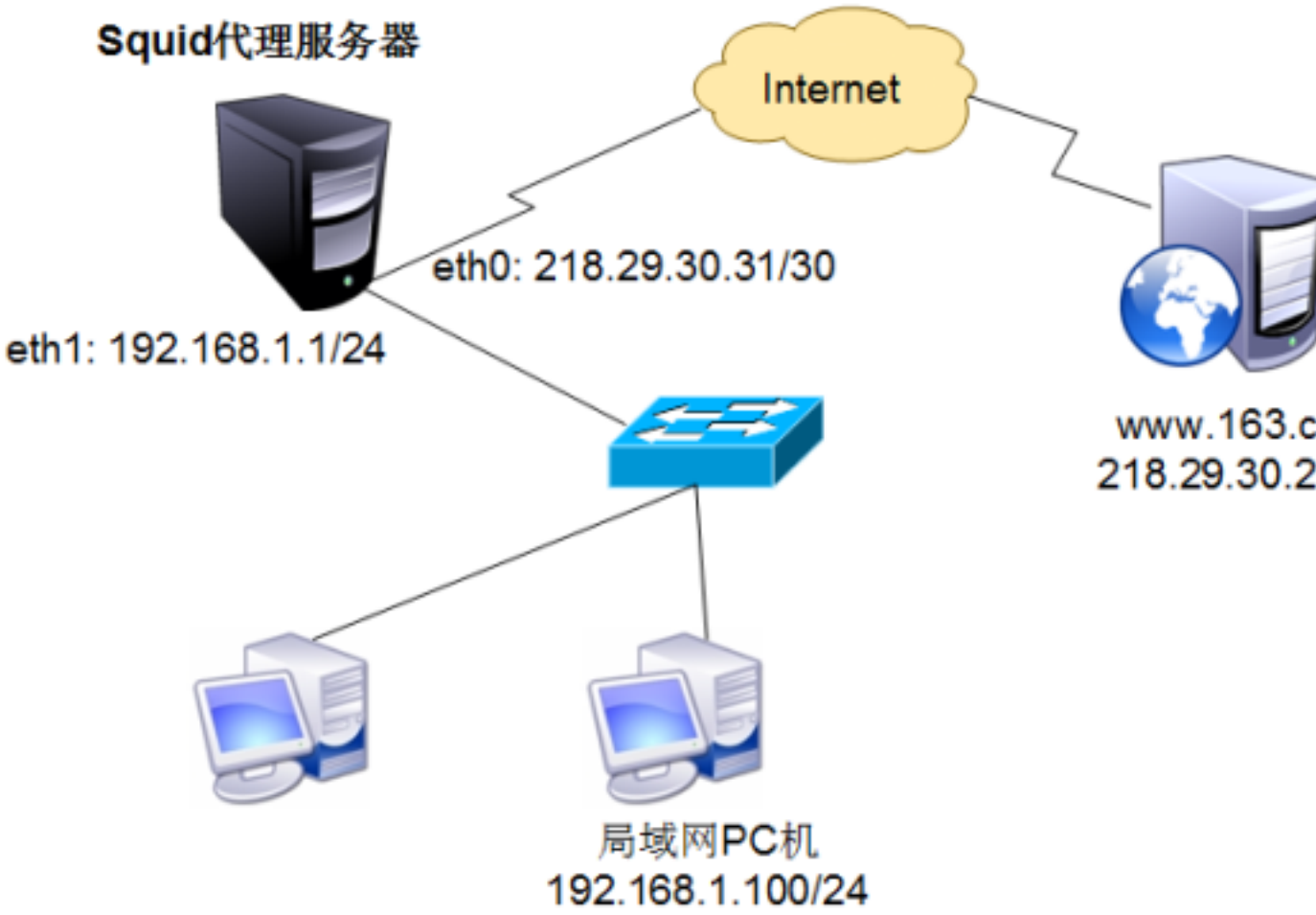
vim /etc/sysctl.conf
net.ipv4.ip_forward = 1
[root@rhel7 ~]# sysctl -p ----立即生效
net.ipv4.ip_forward = 1
2、添加防火墙地址转发策略
iptables -t nat -A POSTROUTING -o eno33554984 -j SNAT --to-source 10.10.10.10 路由后了了要出去找知道的公网IP
或者
iptables -t nat -A POSTROUTING -o eno33554984 -s 172.25.254.0/24 -j MASQUERADE ----不知道公网IP
172.25.254.241 ----客户端
ifconfig eno16777736 172.25.254.241
route add default gw 172.25.254.240

10.10.10.1 -----页面服务器

yum -y install httpd
echo "this is baidu page from 10.10.10.1" > /var/www/html/index.html
systemctl start httpd

测试：
172.25.254.241--->访问10.10.10.1页面
firefox http://10.10.10.1 &

squid 代理服务
正向代理
客户端---->代理服务器----->公网
内网
1 装代理服务器
yum install -y squid
2 客户端的设置 在浏览器上选择代理服务器
编辑--->首选项--->高级--->网络--->连接 设置--->手动添加代理服务器----> 代理服务器 IP 端口3128



DNAT&s反向代理

DNAT　　目的地址转发　　客户端的请求通过DNAT服务转发个真实的服务器

　　客户端：　　DNAT服务器　　真实页面服务器

172.25.254.241 ---> 172.25.254.240 eno16777736 --- eno33554984 10.10.10.10 ----> 10.10.10.1

　　公网--------->通过DNAT服务器<------真室服务器

　　内网的服务器

DNAT 服务端：

1、开启路由转发

　　vim /etc/sysctl.conf

　　net.ipv4.ip_forward = 1

　　sysctl -p　　----即刻生效

2、添加iptables规则

　　iptables -t nat -A PREROUTING -i eno16777736 -p tcp -d 172.25.254.240 --dport 80 -j DNAT --to 10.10.10.1

　　路由前　　进站网卡　　对于客户端来说是目的地址

3、客户端：

　　ifconfig eno16777736 172.25.254.241

4、网页服务器：

　　ifconfig eno16777736 10.10.10.1

systemctl start httpd
echo "this is DNAT page from 10.10.10.1" > /var/www/html/index.html
route add default gw 10.10.10.10 ---找回去的路

5、测试：

172.25.254.241--->172.25.254.240 客户端访问DNAT服务器，然后跳转到Apache服务器

tcpdump -i 网卡 在DNAT服务上监控 进站的网卡

wireshark

反向代理

172.25.254.241 ----172.25.254.240 ----10.10.10.10 ----10.10.10.1

公网IP 内网IP ----10.10.10.2

客户端（公网） 代理服务器（squid ） 内网apache server

1、代理服务器设置反向代理

172.25.254.240

10.10.10.10

vim /etc/squid/squid.conf

http_port 172.25.254.240:80 vhost

cache_peer 10.10.10.1 parent 80 0 originserver weight=5 max-conn=30 --weight 权重 max-conn 最大连接数

cache_peer 10.10.10.2 parent 80 0 originserver weight=5 max-conn=30

systemctl stop httpd ----避免端口冲突

systemctl restart squid

netstat -tnpl |grep squid

tcp 0 0 172.25.254.240:80 :::* LISTEN 19901/(squid-1)

2、后端节点服务器：

10.10.10.1 ----> echo "this is 10.10.10.1 test page" > /var/www/html/index.html

route add default gw 10.10.10.10

systemctl restart httpd

curl http://10.10.10.1 -dump ----获取页面信息返回

this is 10.10.10.1 test page

10.10.10.2 ---->echo "this is 10.10.10.2 test page" > /var/www/html/index.html

route add default gw 10.10.10.10

systemctl restart httpd

curl http://10.10.10.2 -dump

this is 10.10.10.2 test page

3、客户端测试

172.25.254.241---->访问172.25.254.240

curl http://172.25.254.240 -dump

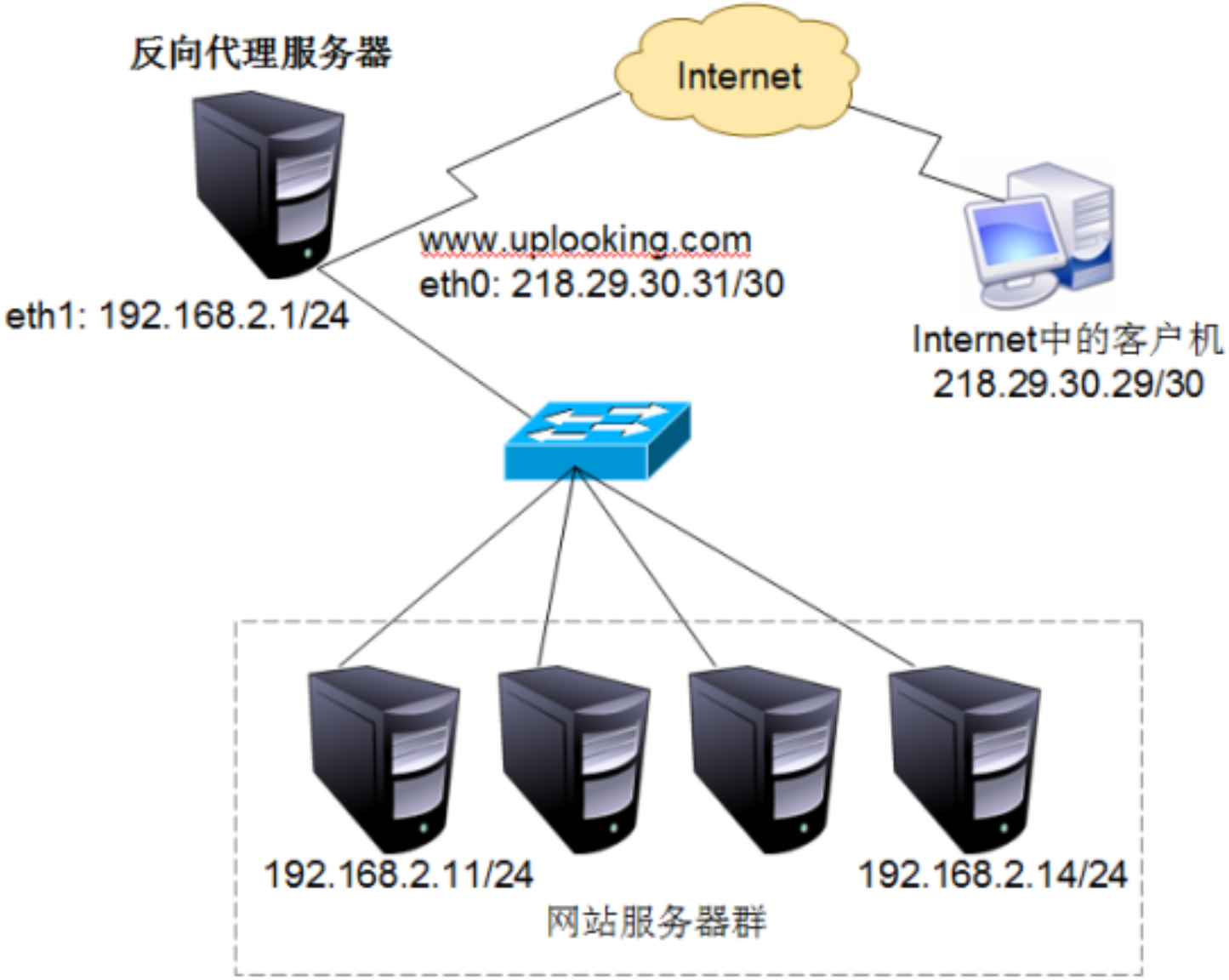
this is 10.10.10.1

模拟故障，将10.1主机的http服务停止，systemctl stop httpd

再测试

curl http://172.25.254.240 -dump

this is 10.10.10.2 test page



透明代理

、透明代理

1、修改代理服务器配置文件 端口转发

vim /etc/squid/squid.conf

http_port 3128 transparent (transparent 透明代理)
cache_dir ufs /var/spool/squid 1000 16 256 --1000表最大容量1000M,16是指squid目录下的一级目录为16个

--256是指16个目录下分别有256个子文件

目录
cache_mem 200 MB --建议给内存的三分之一左右

systemctl restart squid

2、添加防火墙策略

iptables -t nat -A PREROUTING -i eno16777736 -s 172.25.254.0/24 -p tcp --dport 80 -j REDIRECT --to

3128

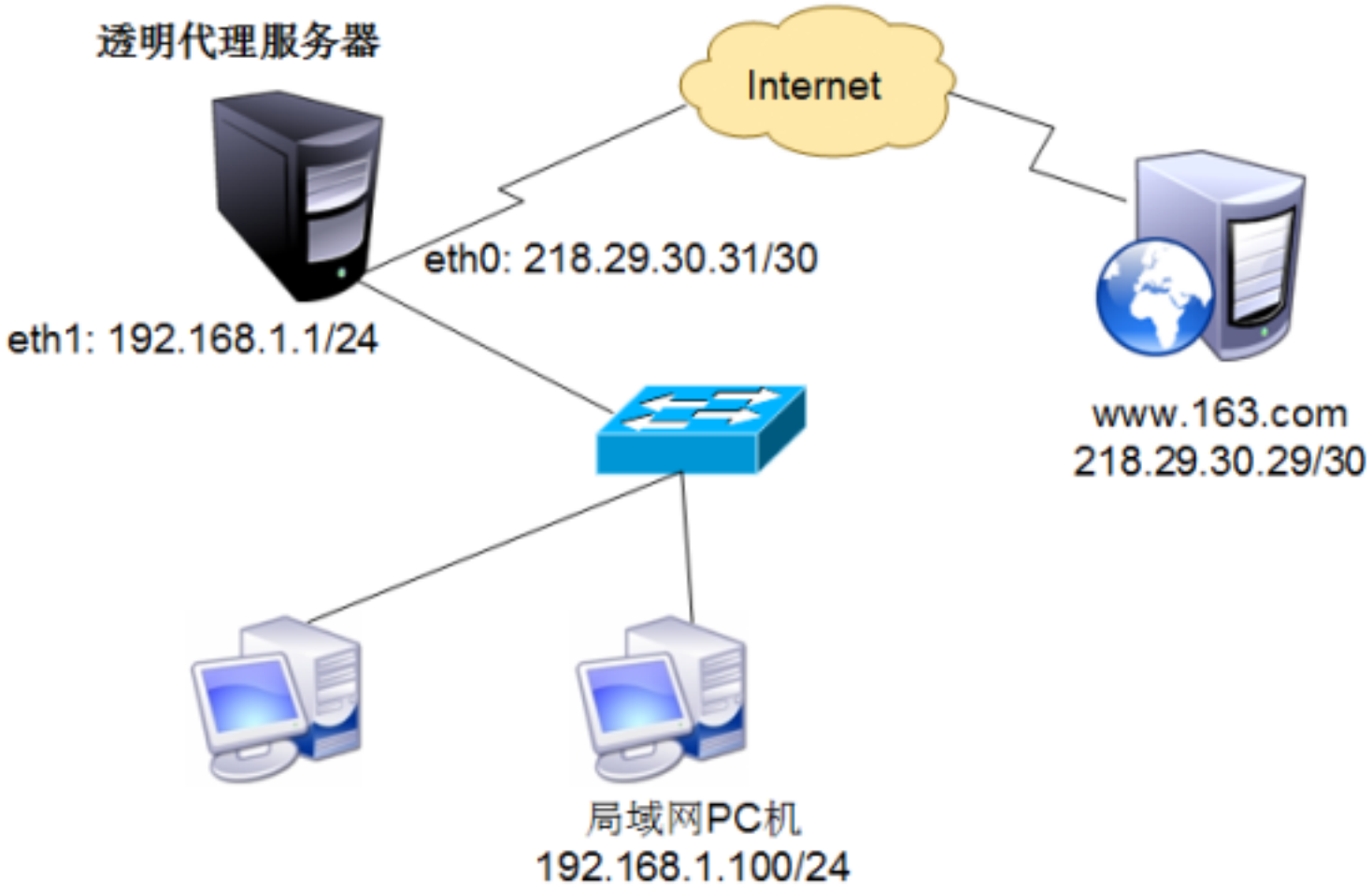
-j REDIRECT 转发
将源自172.25.254.0/24网段的访问80端口的数据包请求全部转发到3128端口处理

echo 1 > /proc/sys/net/ipv4/ip_forward ----路由转发功能开启

2、取消客户端的代理设置
取消手动代理服务器及端口

3、测试：

curl http://10.10.10.1/ --dump ----获取页面信息



firewalld

firewalld防火墙
firewalld-config 图形界面
查看支持zone
firewall-cmd --get-zones
[root@foundation0 Desktop]# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
查看默认zone
[root@foundation0 Desktop]# firewall-cmd --get-default-zone

public
　　查看zone配置信息
[root@foundation0 Desktop]# firewall-cmd --zone=public --list-all
public (default, active)
  interfaces: br0 br1 enp1s0 enp3s0
  sources:
  services: dhcpv6-client ssh
  ports: 80/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

添加允许的服务
[root@foundation0 Desktop]# firewall-cmd --zone=public --add-service=http
success
添加允许的端口
　　[root@foundation0 Desktop]# firewall-cmd --zone=public --add-port=53/tcp
success

删除服务：

[root@foundation0 Desktop]# firewall-cmd --zone=public --remove-service=http
success

删除端口：
[root@foundation0 Desktop]# firewall-cmd --zone=public --remove-port=53/tcp
success

修改默认zone

[root@foundation0 Desktop]# firewall-cmd --set-default-zone=block
success
[root@foundation0 Desktop]# firewall-cmd --get-default-zone
block

重读配置防火墙

firewall-cmd --reload

永久修改策略
　　firewall-cmd --permanent

永久修改不会即刻生效，需要reload重新加载生效
[root@foundation0 Desktop]# firewall-cmd --permanent --add-service=http
success

禁止ping

firewall-cmd --add-icmp-block=echo-request

移除禁ping

firewall-cmd --remove-icmp-block=echo-request

目的地址转发：

firewall-cmd --add-masquerade ----开启转发

所有访问10.10.10.1的22端口的数据包请求指定转发至10.10.10.10的22号端口

firewall-cmd --add-forward-port=port=22:proto=tcp:toaddr=10.10.10.10:toport=22

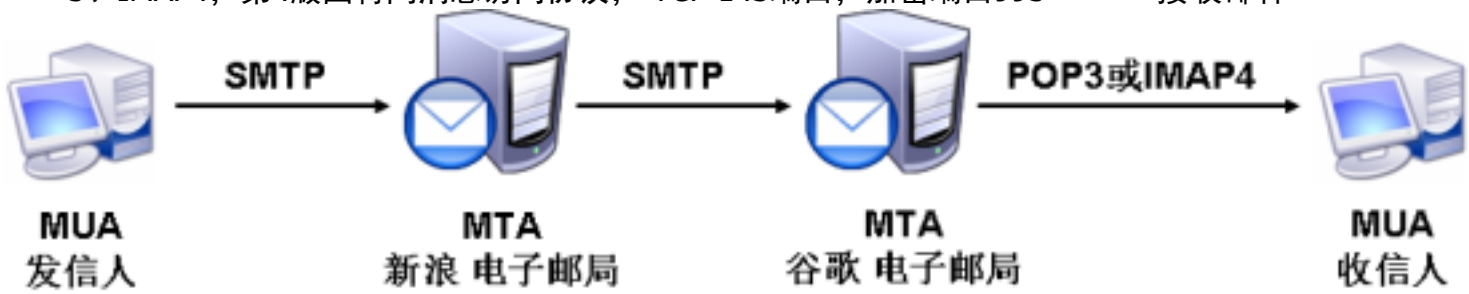
移除端口转发
firewall-cmd --remove-forward-port=port=22:proto=tcp:toaddr=10.10.10.10:toport=22

mail

邮件服务：

MUA---MTA---MTA---MUA
MUA：用户代理 用户代理MUA(Mail User Agent):用于收发邮件。
MTA：邮局代理 邮件传输代理MTA(Mail Transfer Agent):将来自于MUA的邮件转发给指定用户。
邮件投递代理MDA(Mail Delivery Agent):将来自于MTA的邮件保存到本机的收件箱中。

邮件应用协议
1、SMTP， 简单邮件传输协议， TCP 25端口，加密端口465， 发邮件，一般工作在邮件服务器
2、POP3， 第3版邮局协议， TCP 110端口，加密端口995 接收邮件
3、IMAP4，第4版因特网消息访问协议， TCP 143端口，加密端口993 接收邮件



[root@rhel7 ~]# rpm -q postfix
/etc/pam.d/smtp
/etc/postfix/main.cf -----主配置文件
/etc/postfix/master.cf -----管理核心进程配置文件
/usr/bin/mailq -----二进制命令
/usr/sbin/postconf -----配置文件的二进制命令

邮件存放位置
mailbox----/var/mail/username
$HOME/Mailbox

安装邮件服务端：
yum -y install postfix
vim /etc/postfix/main.cf

myhostname = mail.uplooking.com ---主机名
mydomain = uplooking.com ---域
myorigin = $mydomain ---自动补全
inet_interfaces = all ---监听网络
#inet_interfaces = localhost ----注释配置

```
        mydestination = $myhostname, $mydomain, localhost                ---信任域
        mynetworks = 168.100.189.0/28, 127.0.0.0/8 , 172.25.0.0/16       ---信任网络
        home_mailbox = Maildir/                                          ---邮件盒子
[root@rhel7 ~]# systemctl restart postfix
[root@rhel7 ~]# netstat -tnpl |grep :25
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      1977/master
tcp6       0      0 :::25                   :::*                    LISTEN      1977/master

iptables -F
setenforce 0
```

测试客户端：

172.25.254.250

网络发送邮件：

```
                yum -y install telnet
                        telnet 172.25.0.10 25
                        ctrl+]        ----退出telnet

telnet 172.25.0.10 25
Trying 172.25.0.10...
Connected to 172.25.0.10.
Escape character is '^]'.
220 mail.uplooking.com ESMTP Postfix
ehlo localhost                                  ----打招呼
250-mail.uplooking.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from:student                               ----发件人
250 2.1.0 Ok
rcpt to:root                                    ----收件人
250 2.1.5 Ok
data                                            ---数据
354 End data with <CR><LF>.<CR><LF>
subject:this is a test mail                     ----主题
this is mail from student and ,this is test messages        ---正文
do you know

.                                               ---邮件正文的结束
```

结果：

监控邮件服务器的目录：

```
        ls /root/Maildir
                new cur tmp                     new---新的邮件  cur 已经读了的邮件
vim     /root/Maildir/new/XXXXXXXXXXx
Return-Path: <student@uplooking.com>
X-Original-To: root
Delivered-To: root@uplooking.com
Received: from localhost (unknown [172.25.0.250])
        by mail.uplooking.com (Postfix) with ESMTP id 3894C632659E
        for <root>; Wed, 16 Aug 2017 02:42:52 -0400 (EDT)
```

subject:this is a test mail

this is mail from student and ,this is test messages
do you know

whatis wrong

监控邮件日志

vim /var/log/maillog

Aug 16 02:41:46 rhel7 postfix/smtpd[2312]: connect from unknown[172.25.0.250]
Aug 16 02:43:09 rhel7 postfix/smtpd[2312]: 3894C632659E: client=unknown[172.25.0.250]
Aug 16 02:44:27 rhel7 postfix/cleanup[2322]: 3894C632659E: message-id=<>
Aug 16 02:44:27 rhel7 postfix/qmgr[1979]: 3894C632659E: from=<student@uplooking.com>, size=281, nrcpt=1 (queue active)
Aug 16 02:44:27 rhel7 postfix/local[2339]: 3894C632659E: to=<root@uplooking.com>, orig_to=<root>, relay=local, delay=95, delays=95/0.04/0/0.11, dsn=2.0.0, status=sent (delivered to maildir)
Aug 16 02:44:27 rhel7 postfix/qmgr[1979]: 3894C632659E: removed 已近看了

接收邮件服务：

dovecot
[root@rhel7 Maildir]# rpm -ql dovecot
/etc/dovecot
/etc/dovecot/conf.d ----扩展
/etc/dovecot/dovecot.conf ---主配置文件

修改配置文件
vim /etc/dovecot/conf.d/10-auth.conf
10 disable_plaintext_auth = no ----支持明文传输

[root@rhel7 Maildir]# systemctl restart dovecot
[root@rhel7 Maildir]# netstat -tnpl |grep dove
tcp 0 0 0.0.0.0:993 0.0.0.0:* LISTEN 4486/dovecot
tcp 0 0 0.0.0.0:995 0.0.0.0:* LISTEN 4486/dovecot
tcp 0 0 0.0.0.0:110 0.0.0.0:* LISTEN 4486/dovecot ---pop3
tcp 0 0 0.0.0.0:143 0.0.0.0:* LISTEN 4486/dovecot ---imap4

支持网络接收邮件：修改配置信息

vim /etc/dovecot/dovecot.conf

protocols = imap pop3 lmtp
login_trusted_networks = 172.25.0.0/16

vim /etc/dovecot/conf.d/10-mail.conf

mail_location = maildir:~/Maildir
mail_access_groups = mail

[root@rhel7 Maildir]# systemctl restart dovecot

[root@rhel7 Maildir]# telnet 172.25.0.10 110
Trying 172.25.0.10...

```
Connected to 172.25.0.10.
Escape character is '^]'.
+OK Dovecot ready.
user student              ----收邮件的用户
+OK
pass student              ----该用户的密码
+OK Logged in.
list                  ----列出
+OK 1 messages:
1 443
.
retr 1                    ----查看编号1的邮件
+OK 443 octets
Return-Path: <root@uplooking.com>
X-Original-To: student
Delivered-To: student@uplooking.com
Received: from localhost (rhel7-f0.example.com [172.25.0.10])
        by mail.uplooking.com (Postfix) with ESMTP id F39C763277B8
        for <student>; Wed, 16 Aug 2017 03:52:46 -0400 (EDT)
subject:test
Message-Id: <20170816075251.F39C763277B8@mail.uplooking.com>
Date: Wed, 16 Aug 2017 03:52:46 -0400 (EDT)
From: root@uplooking.com

this is test mail
.
quit          ----退出
```

网页邮件：

```
lftp 172.25.254.250
lftp 172.25.254.250:~> cd pub/soft/
lftp 172.25.254.250:/pub/soft> get squirrelmail-webmail-1.4.22.tar.gz
624058 bytes transferred
lftp 172.25.254.250:/pub/soft> exit

[root@rhel7 Maildir]# mv squ* /opt

[root@rhel7 squerrmail]# mkdir /webroot
[root@rhel7 Maildir]#cd /opt/
[root@rhel7 squerrmail]#tar xf squirrelmail-webmail-1.4.22.tar.gz
[root@rhel7 squerrmail]#cp squirrelmail-1.4.22/* /webroot -a
```

创建配置文件：

```
# cd /webroot/config
# cp config_default.php  config.php

# vim /webroot/config/config.php
$domain = 'uplooking.com';
$data_dir = '/webroot/data/';                    --邮件数据
$attachment_dir = '/webroot/attach/';  --邮件附件

 mkdir /webroot/{data,attach} -p
 chown -R apache.apache /webroot/data/ /webroot/attach/
```

vim /etc/httpd/conf/httpd.conf

```
<VirtualHost *:80>
    ServerAdmin  root.example.com
    DocumentRoot  "/webroot"
    ServerName  mail.uplooking.com
    ErrorLog  "/var/log/httpd/www.uplooking.com-error_log"
    CustomLog  "/var/log/httpd/www.uplooking.com-access_log" common
</VirtualHost>
<Directory  "/webroot">
AllowOverride None
Options None
require all granted
</Directory>
```

yum -y install php　　----支持php页面

systemctl restart httpd

测试端：

vim /etc/hosts

172.25.0.10　　mail.uplooking.com

firefox http://mail.uplooking.com

h

EMAIL服务器

电子邮件（英语：electronic mail、简称：e-mail），又称电子邮箱，简称电邮，是指通过互联网进行书写、发送和接收信件，目的是达成发信人和收信人之间的信息交互。

Postfix 是一种电子邮件服务器，它是由任职于IBM华生研究中心（T.J. Watson Research Center）的荷兰籍研究员Wietse Venema为了改良sendmail邮件服务器而产生的。最早在1990年代晚期出现，是一个开放源代码的软件。

默认情况下，postfix监听127.0.0.1的25号端口。可以对本地用户邮箱和对外发送邮件

```
[root@geust02 ~]# ss   -ntupl |grep  master
tcp    LISTEN     0      100    127.0.0.1:25                 *:*                  users:
(("master",pid=1902,fd=13))
tcp    LISTEN     0      100     ::1:25                  :::*                 users:(("master",pid=1902,fd=14))
```

系统命令行邮件客户端mail的使用示例：

```
~]# mail chuyue
Subject: test mail
test mail
Gook luck
```

.
EOT
~]# mail -u chuyue
Heirloom Mail version 12.4 7/29/08. Type ? for help.
"/var/mail/chuyue": 4 messages 4 new
>N 1 Mail Delivery System Tue Jul 9 10:45 76/2567 "Undelivered Mail Returne"
 N 2 Mail Delivery System Tue Jul 9 10:46 75/2574 "Undelivered Mail Returne"
 N 3 Mail Delivery System Tue Jul 9 10:46 77/2671 "Undelivered Mail Returne"
 N 4 root   Wed Jul 24 10:56 19/631   "test mail"
& 4
Message 4:
From root@teacher01.localdomain Wed Jul 24 10:56:13 2013
Return-Path: <root@teacher01.localdomain>
X-Original-To: chuyue
Delivered-To: chuyue@teacher01.localdomain
Date: Wed, 24 Jul 2013 10:56:13 +0800
To: chuyue@teacher01.localdomain
Subject: test mail
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
From: root@teacher01.localdomain (root)
Status: R

test mail
Gook luck

& quit

对外发送也可以， 比如发送给huangdaojin@uplooking.com ，但一般情况下对方服务器会拒收，或者当成垃圾邮件。

配置一个完整的邮件服务器。

需要三个服务器：

dns： 192.168.122.116
mail: 192.168.122.200
client: 192.168.122.1 真机

1. 配置DNS的邮件解析记录

@ in soa localhost. root 1 3H 15M 1W 1D
  ns localhost.
www   A   192.168.122.200
@   MX 10 mail.shangguan.com.
mail A   192.168.122.200
smtp A   192.168.122.200
pop3   CNAME   smtp
imap   CNAME   smtp

~]# nslookup mail.shangguan.com
~]# nslookup smtp.shangguan.com
~]# nslookup pop3.shangguan.com

针对邮件服务器的DNS解析建议配置反向解析。

2. 安装配置smtp发件服务器

~]# yum install postfix
一般已经默认安装了。

可以选择切换MTA，sendmail 或 postfix， RHEL5默认是sendmail RHEL6/7默认是postfix， 以下命令不用执行 。
~]# alternatives --display mta 查看
~]# alternatives --config mta 选择

配置：
[root@mail-server ~]# vim /etc/postfix/main.cf
myhostname = mail.shangguan.com
mydomain = shangguan.com
myorigin = $mydomain
inet_interfaces = all
inet_protocols = ipv4
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mynetworks_style = subnet
mynetworks = 192.168.122.0/24, 127.0.0.0/8
home_mailbox = Maildir/

注：每行文字应该放置行首，而且前面不应有空格或定格字符。以空格或定格字符起首的行会被视为上一行的延续，假如上一行是注释（#），接著的那一行亦会被同样看待。此外，请避免采用内置的注释。

[root@mail-server ~]# systemctl restart postfix

3. 安装配置pop3收件程序

~]# yum install dovecot
~]# vim /etc/dovecot/dovecot.conf
protocols = imap pop3
listen = *

~]# vim /etc/dovecot/conf.d/10-mail.conf
mail_location = maildir:~/Maildir

~]# vim /etc/dovecot/conf.d/10-auth.conf
disable_plaintext_auth = no
auth_mechanisms = plain login

[root@mail-server conf.d]# vim 10-ssl.conf
ssl = no

[root@mail-server conf.d]# systemctl start dovecot
[root@mail-server conf.d]# systemctl enable dovecot

[root@mail-server conf.d]# ss -ntupl |grep dove
tcp LISTEN 0 100 *:110 *:* users:(("dovecot",pid=8408,fd=23))
tcp LISTEN 0 100 *:143 *:* users:(("dovecot",pid=8408,fd=34))

4. 添加邮箱用户

~]# useradd zhangsan
~]# passwd zhangsan
~]# useradd lisi

~]# passwd lisi

5. 在客户端用telnet工具进行测试

~]# vim /etc/resolv.conf
nameserver 192.168.122.116

[root@teacher ~]# yum install telnet

5.1. 发邮件：

```
[root@teacher ~]# telnet mail.shangguan.com smtp
Trying 192.168.122.116...
Connected to mail.shangguan.com.
Escape character is '^]'.
220 mail.shangguan.com ESMTP Postfix
ehlo smtp.shangguan.com
250-mail.shangguan.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from:zhangsan@shangguan.com
250 2.1.0 Ok
rcpt to:lisi@shangguan.com
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
hello lisi ,I am zhangsan.
.
250 2.0.0 Ok: queued as A56C63FF08
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

5.2. 收邮件

```
[root@teacher ~]# telnet mail.shangguan.com pop3
Trying 192.168.122.116...
Connected to mail.shangguan.com.
Escape character is '^]'.
+OK Dovecot ready.
user lisi
+OK
pass 123456
+OK Logged in.
list
+OK 1 messages:
1 288
.
retr 1
+OK 288 octets
Return-Path: <zhangsan@shangguan.com>
X-Original-To: lisi@shangguan.com
Delivered-To: lisi@shangguan.com
Received: from smtp.shangguan.com (unknown [192.168.122.1])
        by mail.shangguan.com (Postfix) with ESMTP id A56C63FF08
```

for <lisi@shangguan.com>; Fri, 13 Sep 2013 02:49:08 +0800 (CST)

hello lisi ,I am zhangsan.
.
quit
+OK Logging out.
Connection closed by foreign host.

PS: 测试是zhangsan@shangguan.com发给lisi@shangguan.com邮件，是属于本域的邮件发送。
如果zhangsan@shangguan.com给2345678@qq.com发邮件，只要邮件服务器能连接外网（dns能解析外网域名地址）即可发送成功。不过，一般情况下qq邮箱会把你发送的邮件当成垃圾邮件拒收（因为你的shangguan.com域名是假的，没有经过公网授权的。IP反解析是不能得到该域名的。）。（测试可能需要在qq邮箱中将zhangsan@shangguan.com添加到白名单）

测试时用telnet命令行来进行的。还可以使用mail mutt命令。也可以用Evolution,foxmail,outlook这样客户端软件来测试。更加直观。
比如演示使用windows foxmail

在Linux环境里面可以在客户端安装Evolution（讲师演示，建议使用)

发邮件的命令行示例：在服务器端
~]# echo "test mail " |mail -s "test" 2241871@qq.com
~]# mutt -s "test postfix" -a spacer.gif -- 8325643@163.com < aaa.txt
~]# mutt -s "test postfix" -c 2241871@qq.com -a spacer.gif -- 8325643@163.com < aaa.txt

还可以在邮件服务器安装、开发WEB界面。（有一些已经开发好的模板，比如openwebmail SquirrelMail ）
讲师演示 SquirrelMail 需要安装邮件服务器上。

```
[root@mail-server ~]# vim /etc/httpd/conf/httpd.conf
<Directory />
    AllowOverride none
    Require all  granted
</Directory>

[root@mail-server squirrelmail]# vim   /etc/httpd/conf.d/squirrelmail.conf
# this section makes squirrelmail use https connections only, for this you
# need to have mod_ssl installed. If you want to use unsecure http
# connections, just remove this section:
#<Directory /usr/share/squirrelmail>
#  RewriteEngine  on
#  RewriteCond    %{HTTPS} !=on
#  RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
#  <IfModule mod_authz_core.c>
#    # Apache 2.4
#    Require all granted
#  </IfModule>
#  <IfModule !mod_authz_core.c>
#    # Apache 2.2
#    Order allow,deny
#    Allow from all
#  </IfModule>
#</Directory>

[root@mail-server squirrelmail]# vim  /etc/squirrelmail/config.php
$squirrelmail_default_language = 'zh_CN';
```

$domain   = 'shangguan.com.';

$default_folder_prefix   = 'Maildir/';

[root@mail-server squirrelmail]# systemctl restart httpd

在浏览器中访问：
http://mail.shangguan.com/webmail/

做邮件服务器需要一定的技术实力（163，qq,gmail提供企业、个人邮箱的服务），难点在于反垃圾邮件和防病毒，多用户的高性能，大存储。

练习：
1. 搭建一个DNS服务器用来进行MAIL解析；
2. 搭建一个邮件服务器， 可以收发邮件，用户邮箱是zhangsan@shangguan.com
3. 搭建另一个邮件服务器， 可以收发邮件，用户邮箱是lisi@lovelinux.com
4. windows 客户端， 安装foxmail进行收发邮件 或 Linux客户端，安装Evolution收发邮件。
5. zhangsan@shangguan.com 可以在客户端给lisi@lovelinux.com发送邮件，并lisi在客户端接收成功。
6. lisi@lovelinux.com 可以在客户端给zhangsan@shangguan.com发送邮件，并zhangsan在客户端接收成功。

扩展：
尝试给mail.shangguan.com搭建web界面的邮箱服务。

ntp

ntp 时间服务器

装

yum install -y ntp

配置

1. 什么都不配置 网络通就行

2.网络不通

#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst

restrict default nomodify
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /var/lib/ntp/dirft

启动

systemctl restart ntpd
systemctl enable ntpd

测试

客户端装

yum install -y ntpdate

在客户端 配置周期任务

vim /etc/crontable

0 */1 * * * /usr/sbin/ntpdate 192.168.10.254