

1. 常用的网络工具:

1.1 IP 命令的使用

查看 IP 地址

```
[root@svr7 ~]# ip address show
```

添加 IP 地址

```
[root@svr7 ~]# ip address add 192.168.8.1/24 dev ens33
```

#会有8.1的IP, 但是是临时的, 只能通过ip address show查看, ifconfig查看不到

```
[root@svr7 ~]# ip addr show
```

```
[root@svr7 ~]# ping 192.168.8.1 #可以ping通
```

如果想要每次开机生效, 可以写/etc/rc.d/rc.local 文件, 给个执行的权限

```
[root@svr7 ~]# vim /etc/rc.d/rc.local ip address add 192.168.8.1/24 dev ens33
```

添加路由

```
[root@svr7 ~]# ip route add 10.0.0.0/24 via 192.168.8.100 dev eth0 #via是下一跳
```

```
[root@svr7 ~]# ip route show #查看路由表
```

删除路由

```
[root@svr7 ~]# ip route del 10.0.0.0/24
```

```
[root@svr7 ~]# ip route show
```

1.2 ss 与 netstat

-a 显示所有端口的信息

-n 以数字格式显示端口号

-t 显示 TCP 连接的端口

-u 显示 UDP 连接的端口

-l 显示服务正在监听的端口信息

-p 显示监听端口的服务名称是什么（也就是程序名称）

```
[root@svr7 ~]# netstat -anptu
[root@svr7 ~]# ss -anptu
[root@svr7 ~]# netstat -anptu | grep :22
[root@svr7 ~]# ss -anptu | grep :22
```

1.3 ping 命令

常用选项：-c 测试包个数

```
[root@svr7 ~]# ping -c 2 192.168.4.207 #ping2次结束
```

2. 源码编译安装：

2.1 源码编译安装的优势

RPM 软件包：rpm -ivh yum -y install

源码包---->开发工具----->可以执行的程序----->运行安装

优势：

- 1) 灵活度较高，自定义较高，可以指定安装位置
- 2) 获得软件的最新版,及时修复 bug
- 3) 软件功能可按需选择/定制,有更多软件可供选择
- 4) 源码包适用各种平台

2.2 源码编译安装的基本实现过程



2.2 源码编译安装的步骤

步骤一: 安装开发工具 **gcc** 与 **make**

```
[root@svr7 ~]# yum -y install gcc make #可能会报错, 报错参考下面步骤修复
[root@svr7 ~]# mount /dev/cdrom /dvd/
[root@svr7 ~]# vim /etc/yum.repos.d/dvd.repo
[dvd]
name=CentOS7.5
baseurl=file:///dvd
enabled=1
gpgcheck=0
[root@svr7 ~]# yum -y install gcc make
```

查看是否安装成功

```
[root@svr7 ~]# rpm -q gcc make
gcc-4.8.5-28.el7.x86_64
```

```
make-3.82-23.el7.x86_64
```

步骤二: tar 解包,释放源代码至指定目录

```
[root@svr7 ~]# tar -xf tools.tar.gz -C /opt
[root@svr7 ~]# ls /opt/tools/
[root@svr7 ~]# tar -xf /opt/tools/inotify-tools-3.13.tar.gz -C /opt/
[root@svr7 ~]# ls /opt/
[root@svr7 ~]# cd /opt/inotify-tools-3.13/
```

步骤三: ./configure 配置,指定安装目录/功能模块等选项, 检测系统是否安装了 gcc 软件

--prefix=安装路径

```
[root@svr7 inotify-tools-3.13]# ./configure --prefix=/opt/myrpm #指定安装目录, 目录可以不存在
```

常见的错误提示, 没有安装 gcc

```
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details.
```

步骤四: make 编译,生成可执行的二进制程序文件

```
[root@svr7 inotify-tools-3.13]# make
```

步骤五: make install 安装,将编译好的文件复制到安装目录

```
[root@svr7 inotify-tools-3.13]# make install
[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/
[root@svr7 inotify-tools-3.13]# ls /opt/myrpm/bin/
```

3. 自定义 yum 仓库:

3.1 yum 仓库的定义

自定义 Yum 仓库: 将自己准备好的软件包,进行 Yum 机制的管理

3.2 自定义 yum 仓库的配置

步骤一: 准备软件包, 这个已经提前准备好, 在 tools 压缩包里

```
[root@svr7 ~]# ls /opt/tools/other/ #注: tools包释放到哪个目录就写哪里
```

步骤二: 通过命令生成仓库数据文件

```
[root@svr7 ~]# createrepo /opt/tools/other/  
[root@svr7 ~]# ls /opt/tools/other/      #再次查看有repodata目录
```

步骤三: 书写客户端配置文件

```
[root@svr7 ~]# vim /etc/yum.repos.d/dvd.repo  
[dvd]  
name=CentOS7.5  
baseurl=file:///dvd  
enabled=1  
gpgcheck=0  
[nsd]  
name=myrpm  
baseurl=file:///opt/tools/other  
enabled=1  
gpgcheck=0  
[root@svr7 ~]# yum clean all  
[root@svr7 ~]# yum repolist
```

步骤四: 安装软件包

```
[root@svr7 ~]# yum -y install sl
[root@svr7 ~]# sl
[root@svr7 /]# yum -y install cmatrix
[root@svr7 /]# cmatrix
[root@svr7 /]# yum -y install oneko
[root@svr7 /]# oneko
[root@svr7 /]# oneko & #将程序放入后台运行
[root@svr7 /]# killall oneko #杀死所有oneko程序
[root@svr7 /]# yum -y install boxes
[root@svr7 /]# echo "I am dog" | boxes -d dog -a c
```

1. SELinux:

1.1 SELinux 概述

Security-Enhanced Linux

- 美国 NSA 国家安全局主导开发,一套增强 Linux 系统安全的强制访问控制体系
- 集成到 Linux 内核(2.6 及以上)中运行
- RHEL7 基于 SELinux 体系针对用户、进程、目录和文件 提供了预设的保护策略,以及管理工具

1.2 SELinux 的运行模式及切换

1.2.1 SELinux 的运行模式

- enforcing(强制)
- permissive(宽松)
- disabled(彻底禁用)

任何模式再切换到 disabled(彻底禁用)模式, 都必须经历重起系统

1.2.2 SELinux 的运行模式切换

- 临时切换: `setenforce 1|0`
- 固定配置: `/etc/selinux/config` 文件 #不影响当前 SELinux 状态

临时配置

<code>[root@localhost ~]# getenforce</code>	#查看SELinux当前的状态 Enforcing
<code>[root@localhost ~]# setenforce 0</code>	#更改当前模式, 0是宽松状态
<code>[root@localhost ~]# getenforce</code>	
Permissive	
<code>[root@localhost ~]# setenforce 1</code>	#更改当前模式, 1是强制状态

```
[root@localhost ~]# getenforce
```

Enforcing

永久更改

```
[root@localhost ~]# vim /etc/selinux/config      #彻底禁用模式
.....
SELINUX=disabled
.....
[root@localhost ~]# reboot
```

重启系统查看 SELinux 状态

```
[root@localhost ~]# getenforce
Disabled
```


1. C/S 服务架构:

1.1 什么服务器

能够为其他计算机提供服务的更高级的电脑

1.2 服务器类型

--机架式

--塔式

--机柜式

--刀片式

注：1U 服务器尺寸：规定的尺寸是服务器的宽（48.26cm=19 英寸）与高（4.445cm 的倍数）

1U=4.445cm

1.3 典型服务模式

C/S,Client/Server 架构

— 由服务器提供资源或某种功能

— 客户机使用资源或功能

2. 构建 WEB 服务&FTP 服务:

2.1 构建 Web 服务（在虚拟机 A 上操作）

2.1.1 安装 httpd 软件包

```
[root@A ~]# yum -y install httpd #注：安装失败检测yum源是否可用
```

```
[root@A ~]# rpm -q httpd
```

2.1.2 启动 httpd 服务

```
[root@A ~]# systemctl start httpd
```

2.1.3 书写页面文件，测试

默认存放网页文件的路径：/var/www/html

默认网页文件的名称：index.html

```
[root@A ~]# vim /var/www/html/index.html
```

I am king.

测试

```
[root@A ~]# curl http://192.168.4.7
```

2.2 搭建 FTP 服务（在虚拟机 A 上操作）

2.2.1 安装 vsftpd 软件包

```
[root@A~]# yum -y install vsftpd #注：安装失败检测yum源是否可用
```

```
[root@A ~]# rpm -q vsftpd
```

2.2.2 启动 httpd 服务

```
[root@A ~]# systemctl start vsftpd
```

2.2.3 测试

默认共享数据目录：/var/ftp

```
[root@A ~]# touch /var/ftp/a.txt #创建测试文件
```

```
[root@A ~]# curl ftp://192.168.4.7
```

3. 防火墙简介：

3.1 防火墙简介

防火墙分为硬件防火墙，软件防火墙

作用：隔离，进行过滤所有入站请求

3.2 firewalld 服务基础

- 管理工具:firewall-cmd、firewall-config (图形工具)

预设安全区域

- 根据所在的网络场所区分,预设保护规则集
 - public: 仅允许访问本机的 sshd、ping、dhcp 服务
 - trusted: 允许任何访问
 - block: 阻塞任何来访请求 (明确拒绝, 有回应客户端)
 - drop: 丢弃任何来访的数据包 (直接丢弃, 没有回应客户端), 节省服务器资源

防火墙判定规则: (进入哪一个区域)

1.首先查看, 客户端数据包中源 IP 地址, 然后查看自己所有区域规则, 那个区域有该源 IP 地址的规则, 则进入该区域

2.进入默认区域 (public)

3.3 防火墙默认区域的修改

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --get-default-zone    #查看默认区域
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# ping 192.168.4.10    #可以通信
[root@svr7 ~]# curl 192.168.4.10    #拒绝访问
[root@svr7 ~]# curl ftp://192.168.4.10    #拒绝访问
```

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --set-default-zone=trusted #修改默认区域为trusted
[root@A ~]# firewall-cmd --get-default-zone
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# curl ftp://192.168.4.10 #可以访问
```

```
[root@svr7 ~]# ping 192.168.4.10 #可以通信
```

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --set-default-zone=block #修改默认区域为block
```

```
[root@A ~]# firewall-cmd --get-default-zone
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# ping 192.168.4.10 #不可以通信, 有回应
```

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --set-default-zone=drop #修改默认区域为drop
```

```
[root@A ~]# firewall-cmd --get-default-zone
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# ping 192.168.4.10 #不可以通信, 没有回应
```

1. 环境准备：

1.1 构建 Web 服务（在虚拟机 A 上操作）

1.1.1 安装 httpd 软件包

```
[root@A ~]# yum -y install httpd #注：安装失败检测yum源是否可用  
[root@A ~]# rpm -q httpd
```

1.1.2 启动 httpd 服务

```
[root@A ~]# systemctl start httpd
```

1.1.3 书写页面文件，测试

默认存放网页文件的路径：/var/www/html

默认网页文件的名称：index.html

```
[root@A ~]# vim /var/www/html/index.html  
I am king.  
测试  
[root@A ~]# curl http://192.168.4.7
```

1.2 搭建 FTP 服务（在虚拟机 A 上操作）

1.2.1 安装 vsftpd 软件包

```
[root@A ~]# yum -y install vsftpd #注：安装失败检测yum源是否可用  
[root@A ~]# rpm -q vsftpd
```

1.2.2 启动 httpd 服务

```
[root@A ~]# systemctl start vsftpd
```

1.2.3 测试

默认共享数据目录：/var/ftp

```
[root@A ~]# touch /var/ftp/a.txt #创建测试文件  
[root@A ~]# curl ftp://192.168.4.7
```

2. 防火墙简介:

2.1 防火墙简介

防火墙分为硬件防火墙，软件防火墙

作用：隔离，进行过滤所有入站请求

2.2 firewalld 服务基础

- 管理工具:firewall-cmd、firewall-config (图形工具)

预设安全区域

- 根据所在的网络场所区分,预设保护规则集

– public: 仅允许访问本机的 sshd、ping、dhcp 服务

– trusted: 允许任何访问

– block: 阻塞任何来访请求 (明确拒绝, 有回应客户端)

– drop: 丢弃任何来访的数据包 (直接丢弃, 没有回应客户端), 节省服务器资源

防火墙判定规则: (进入哪一个区域)

1.首先查看, 客户端数据包中源 IP 地址, 然后查看自己所有区域规则, 那个区域有该源 IP 地址的规则, 则进入该区域

2.进入默认区域 (public)

2.3 防火墙默认区域的修改

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --get-default-zone #查看默认区域
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# ping 192.168.4.10      #可以通信
[root@svr7 ~]# curl 192.168.4.10      #拒绝访问
[root@svr7 ~]# curl ftp://192.168.4.10  #拒绝访问
```

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --set-default-zone=trusted #修改默认区域为trusted
[root@A ~]# firewall-cmd --get-default-zone
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# curl ftp://192.168.4.10      #可以访问
[root@svr7 ~]# ping 192.168.4.10            #可以通信
```

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --set-default-zone=block #修改默认区域为block
[root@A ~]# firewall-cmd --get-default-zone
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# ping 192.168.4.10      #不可以通信, 有回应
```

虚拟机 A 操作:

```
[root@A ~]# firewall-cmd --set-default-zone=drop #修改默认区域为drop
[root@A ~]# firewall-cmd --get-default-zone
```

虚拟机 B (svr7) 测试:

```
[root@svr7 ~]# ping 192.168.4.10      #不可以通信, 没有回应
```

2.4 防火墙的策略管理

2.4.1 互联网常见的协议

http: 超文本传输协议 默认端口: 80

https: 安全的超文本传输协议	默认端口: 443
ftp: 文件传输协议	默认端口: 21
tftp: 简单文件传输协议	默认端口: 69
DNS: 域名解析协议	默认端口: 53
telnet: 远程管理协议	默认端口: 23
smtp: 邮件协议 (发邮件)	默认端口: 25
pop3: 邮件协议 (收邮件)	默认端口: 110
snmp: 简单的网络管理协议	默认端口: 161

2.4.2 封网段, 开服务

虚拟机 A 查看区域为 public 区域

```
[root@A ~]# firewall-cmd --get-default-zone  
public
```

虚拟机 B (svr7) 主机测试访问 http 和 ftp, 访问失败

```
[root@svr7 ~]# curl ftp://192.168.4.10  
[root@svr7 ~]# curl 192.168.4.10
```

设置允许 http 协议通过 public 区域

```
[root@A ~]# firewall-cmd --zone=public --add-service=http
```

查看区域策略

```
[root@A ~]# firewall-cmd --zone=public --list-all
```

虚拟机 B (svr7) 主机测试访问

```
[root@svr7 ~]# curl http://192.168.4.10    #成功  
[root@svr7 ~]# curl ftp://192.168.4.10    #失败
```

设置允许 ftp 协议通过 public 区域


```
[root@A ~]# firewall-cmd --zone=public --add-service=ftp  
[root@A ~]# firewall-cmd --zone=public --list-all
```

虚拟机 B (svr7) 主机测试访问

```
[root@svr7 ~]# curl ftp://192.168.4.10
```

测试若是重启机器或者重新加载防火墙，配置会失效

```
[root@A ~]# reboot
```

需要永久性配置：

永久(--permanent)

```
[root@A ~]# firewall-cmd --permanent --zone=public --add-service=http  
[root@A ~]# firewall-cmd --permanent --zone=public --add-service=ftp  
[root@A ~]# firewall-cmd --reload      #重新加载配置文件  
[root@A ~]# firewall-cmd --zone=public --list-all
```

单独拒绝虚拟机 pc207 (192.168.4.207) 进行访问本机所有服务

虚拟机 A 配置拒绝策略

```
[root@A ~]# firewall-cmd --zone=block --add-source=192.168.4.7
```

虚拟机 B (svr7) 主机测试访问

```
[root@svr7 ~]# curl ftp://192.168.4.10 #失败  
[root@svr7 ~]# curl 192.168.4.10 #失败
```

其他主机可以访问成功

虚拟机 A 删除规则

```
[root@A ~]# firewall-cmd --zone=block --remove-source=192.168.4.7
```

1. 网络 yum 源实战:

1.1 yum 仓库概述

1.1.1 作为 yum 软件源的目录需要准备的内容:

- 1) 大量的 .rpm 软件安装包文件
- 2) 针对这些软件包的 repodata/ 仓库档案

--repodata/ 仓库档案提供的数据:

filelists.xml.gz: 提供所有软件包的文件安装清单

primary.xml.gz: 提供所有软件包的基本/主要信息

other.xml.gz: 提供所有软件包的其他信息

repomd.xml: 提供上述档案数据文件

.xml.gz 的下载和校验信息

1.1.2 验证仓库档案的重要性, 删除 repodata 文件

```
[root@svr7 ~]# mkdir -p /dvd/nsd
[root@svr7 ~]# mount /dev/cdrom /dvd/nsd
[root@svr7 ~]# vim /etc/yum.repos.d/mnt.repo
[dvd]
name=dvd
baseurl=file:///dvd
enabled=1
gpgcheck=0
[root@svr7 ~]# yum clean all
[root@svr7 ~]# yum repolist          #无软件包
```

修改 yum 客户端配置文件, 将服务端 URL 地址修改为/dvd/nsd

```
[root@svr7 ~]# vim /etc/yum.repos.d/mnt.repo
[dvd]
name=dvd
baseurl=file:///dvd/nsd
enabled=1
gpgcheck=0
[root@svr7 ~]# yum clean all
[root@svr7 other]# yum repolist    #有软件包
```

1.2 部署网络 yum 源

1.2.1 服务端：利用 Web 服务或 FTP 服务共享光盘所有内容

1.2.2 利用 web（HTTP）服务共享，默认共享位置：[/var/www/html/](#)

服务端svr7操作

```
[root@svr7 ~]# yum -y install httpd
[root@svr7 ~]# systemctl start httpd
[root@svr7 ~]# systemctl status httpd #查看服务运行状态
[root@svr7 ~]# mkdir /var/www/html/dvd
[root@svr7 ~]# mount /dev/cdrom /var/www/html/dvd
[root@svr7 ~]# ls /var/www/html/dvd #查看是否有光盘内容
[root@svr7 ~]# firefox http://192.168.4.7/dvd #访问测试
```

客户端pc207操作

```
[root@pc207 ~]# vim /etc/yum.repos.d/dvd.repo
[dvd]
name=CentOS7.5
baseurl=http://192.168.4.7/dvd
enabled=1
gpgcheck=0
```

```
[root@pc207 ~]# yum clean all
[root@pc207 ~]# yum repolist
[root@pc207 ~]# yum -y install unzip
```

1.2.3 利用 FTP 服务共享，默认共享位置： /var/ftp

服务端svr7操作

```
[root@svr7 ~]# rpm -q vsftpd
[root@svr7 ~]# systemctl start vsftpd
[root@svr7 ~]# systemctl status vsftpd 查看服务运行状态
[root@svr7 tools]# mkdir /var/ftp/dvd
[root@svr7 tools]# mount /dev/cdrom /var/ftp/dvd
[root@svr7 tools]# firefox ftp://192.168.4.7/dvd #访问测试
```

客户端pc207操作

```
[root@pc207 ~]# vim /etc/yum.repos.d/dvd.repo
[dvd]
name=CentOS7.5
baseurl=ftp://192.168.4.7/dvd
enabled=1
gpgcheck=0
[root@pc207 ~]# yum clean all
[root@pc207 ~]# yum repolist
```

1. NTP 时间同步:

1.1 NTP 协议概述

- Network Time Protocol
 - NTP 服务器为客户机提供标准时间
 - NTP 客户机需要与 NTP 服务器保持沟通
- NTP 时间服务
 - 软件包: chrony
 - 配置文件: /etc/chrony.conf
 - 系统服务: chronyd

1.2 部署 NTP 时间服务

服务端svr7操作

```
[root@svr7 ~]# rpm -q chrony
[root@svr7 ~]# rpm -qc chrony          #查看配置相关文件
[root@svr7 ~]# vim /etc/chrony.conf    #数字为行号
3 server 0.centos.pool.ntp.org iburst
4 #server 1.centos.pool.ntp.org iburst
5 #server 2.centos.pool.ntp.org iburst
6 #server 3.centos.pool.ntp.org iburst
26 allow 0.0.0.0/0                     #允许所有网段访问
29 local stratum 10                    #设置NTP服务器的层数为10层
[root@svr7 ~]# systemctl restart chronyd
[root@svr7 ~]# systemctl enable chronyd
```

客户端pc207操作

```
[root@pc207 ~]# vim /etc/chrony.conf
```

```
3 server 192.168.4.7 iburst          #指定时间同步地址4.7
4 #server 1.centos.pool.ntp.org iburst      #开头加上#号，变成注释
5 #server 2.centos.pool.ntp.org iburst      #开头加上#号，变成注释
6 #server 3.centos.pool.ntp.org iburst      #开头加上#号，变成注释

[root@pc207 ~]# systemctl restart chronyd
[root@pc207 ~]# systemctl enable chronyd
[root@pc207 ~]# chronyc sources -v #出现*为成功

注：如果失败，请关闭防火墙和selinux，svr7和pc207都关闭
```

2. 实验案例：

服务端是 svr7，客户端为 pc207，完成以下案例：

案例 1：构建网络 yum

利用 FTP 服务实现 yum 源提供

- 1、svr7 构建 vsftpd 服务
- 2、利用 vsftpd 服务提供如下内容：
 - 1) Centos7 光盘内容
 - 2) 自定义 yum 仓库内容
- 3、利用 pc207 进行测试

案例 2：高级远程管理

- 1、实现 svr7 远程管理 pc207，无密码验证
- 2、将 svr7 的/home 目录拷贝到 pc207 的/opt 目录下
- 3、将 svr7 的/etc/passwd 文件拷贝到 tom 用户的家目录下，以用户 tom 的密码验证（用户 tom 密码为 redhat）

1. ISCSI 服务基础:

1.1 ISCSI 概述

- Internet **SCSI**, 网际 SCSI 接口
 - 一种基于 C/S 架构的虚拟磁盘技术
 - 服务器提供磁盘空间, 客户机连接并当成本地磁盘使用

1.2 ISCSI 磁盘的构成

- backstore, 后端存储
 - 对应到服务端提供实际存储空间的设备, 需要起一个管理名称
- target, 磁盘组
 - 是客户端的访问目标, 作为一个框架, 由多个 lun 组成
- lun, 逻辑单元
 - 每一个 lun 需要关联到某一个后端存储设备, 在客户端会视为一块虚拟硬盘

1.3 实验环境准备

虚拟机 **svr7**, 虚拟机 **pc207** 的防火墙默认区域设置为 **trusted** 或者关闭, 关闭 SELinux

```
[root@svr7 ~]# firewall-cmd --set-default-zone=trusted
[root@svr7 ~]# setenforce 0
[root@pc207 ~]# firewall-cmd --set-default-zone=trusted
[root@pc207 ~]# setenforce 0
```

svr7 和 **pc207** 的 yum 仓库可以正常使用

```
[root@svr7 ~]# yum clean all
[root@svr7 ~]# yum repolist
```

发布 ISCSI 磁盘

为虚拟机 **svr7** 添加一块 10G 硬盘, 利用 **/dev/sdb** 划分一个主分区, 大小为 5G

```
[root@svr7 ~]# fdisk /dev/sdb
```


安装服务端软件包 targetcli

```
[root@svr7 ~]# yum -y install targetcli
```

运行 targetcli 命令进行配置

```
[root@svr7 ~]# targetcli
```

```
/> ls
```

1、创建后端存储

```
/> backstores/block create dev=/dev/sdb1 name=nsd
```

```
/> ls
```

2、创建磁盘组target

```
/> iscsi/ create iqn.2019-09.cn.tedu:server
```

```
/> ls
```

3、进行lun关联

```
/> iscsi/iqn.2019-09.cn.tedu:server/tpg1/luns create /backstores/block/nsd
```

```
/> ls
```

4、设置访问控制(acl),设置 客户端声称的名字

```
/> iscsi/iqn.2019-09.cn.tedu:server/tpg1/acls create iqn.2019-09.cn.tedu:client
```

```
/> ls /> exit
```

重启服务并加入开机自启

```
[root@svr7 ~]# systemctl restart target.service
```

```
[root@svr7 ~]# systemctl enable target.service
```

1.5 ISCSI 客户端

访问 ISCSI 磁盘

虚拟机 pc207 操作:

1、安装客户端软件: iscsi-initiator-utils

```
[root@pc207 ~]# yum -y install iscsi-initiator-utils
```

```
[root@pc207 ~]# rpm -q iscsi-initiator-utils
```

2、修改配置文件，指定客户端声称的名字

```
[root@pc207 ~]# vim /etc/iscsi/initiatorname.iscsi  
InitiatorName=iqn.2019-09.cn.tedu:client
```

3、重起 iscsid 服务,仅仅是刷新客户端声称的名字

```
[root@pc207 ~]# systemctl restart iscsid
```

4、利用命令发现服务端共享存储

```
[root@svr7 ~]# man iscsiadm      #iscsiadm man帮助查看  
全文查找/example 按n项下跳转匹配  
[root@pc207 ~]# iscsiadm --mode discoverydb --type sendtargets --portal \  \  
192.168.4.7 --discover 192.168.4.7:3260,1 iqn.2019-09.cn.tedu:server
```

注：\为换行符

5、重启 iscsi 服务,使用共享存储

```
[root@pc207 ~]# systemctl restart iscsi  
[root@pc207 ~]# systemctl enable iscsi  
[root@pc207 ~]# lsblk
```

1.6 总结

服务端:

- 1、修改防火墙默认区域
 - 2、划分新的分区
 - 3、安装服务端 targetcli 软件
 - 4、运行 targetcli 进行配置
- 1) 建立后端存储

- 2) 建立磁盘组 target
- 3) 进行关联 lun
- 4) 设置 ACL, 书写客户端声称的名字
- 5) 启用相应 IP 地址及端口 (可以不做, 默认监听所有)
- 5、重起 target 服务

客户端:

- 1、安装客户端软件包 iscsi-initiator-utils
- 2、修改配置文件,设置客户端声称的名字 `vim /etc/iscsi/initiatorname.iscsi`
- 3、重启 iscsid 进行刷新
- 4、执行发现共享存储命令
- 5、 重启 iscsi 客户端服务, 使用共享存储

1.7 案例练习

配置 iSCSI 服务端

配置 svr7 提供 iSCSI 服务, 磁盘名为 `iqn.2016-02.com.example:svr7`, 服务端口为 3260, 使用 store 作其后端卷, 其大小为 3GiB

配置 iSCSI 客户端

配置 pc207 使其能连接上 svr7 提供的 `iqn.2016-02.com.example:svr7`, iSCSI 设备在系统启动的期间自动加载, 块设备 iSCSI 上包含一个大小为 2100MiB 的分区, 并格式化为 ext4 文件系统 此分区挂载在 `/mnt/data` 上, 同时在系统启动的期间自动挂载

一、Web 基础服务

P1 独立 web 主机

Web 通信基本概念

- 基于 B/S (Browser/Server)架构的网页服务
 - 服务端提供网页
 - 浏览器下载并显示网页
- Hyper Text Markup Language (html) ,超文本标记语言
- Hyper Text Transfer Protocol (http) ,超文本传输协议

Centos7 中的 Web 服务

- 软件包:httpd
- 系统服务:httpd
- 提供的默认配置
 - Listen: 监听地址:端口(80)
 - ServerName: 本站点注册的 DNS 名称(空缺)
 - DocumentRoot: 网页文件根目录(/var/www/html)
 - DirectoryIndex: 起始页/首页文件名(index.html)

虚拟机 **svr7** 作为服务端，搭建 **web** 服务器

1、安装软件包

```
[root@svr7 ~]# yum -y install httpd
```

2、书写页面内容，重启服务

```
[root@svr7 ~]# echo xixhaha > /var/www/html/index.html
[root@svr7 ~]# systemctl restart httpd
```

pc207 作为客户端测试，成功

```
[root@pc207 ~]# curl 192.168.4.7
```

3、修改配置文件 `/etc/httpd/conf/httpd.conf`，改变网页文件存放路径

```
[root@svr7 ~]# vim /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/myweb"           #改变网页文件存放路径
[root@svr7 ~]# mkdir /var/www/myweb
[root@svr7 ~]# echo wo shi myweb > /var/www/myweb/index.html
[root@svr7 ~]# systemctl restart httpd
```

pc207 作为客户端测试，成功

```
[root@pc207 ~]# curl 192.168.4.7
```

虚拟机 svr7 作为服务端，更改 httpd 监听端口为 8080

```
[root@svr7 ~]# vim /etc/httpd/conf/httpd.conf
Listen 8080
[root@svr7 ~]# systemctl restart httpd
[root@svr7 ~]# ss -antlp | grep 8080           #监听的端口为8080
```

测试 pc207 测试，直接访问会失败，需要加端口访问

```
[root@pc207 ~]# curl 192.168.4.7           #失败
[root@pc207 ~]# curl 192.168.4.7:8080       #成功
```

请还原 http 服务的 80 端口

```
[root@svr7 ~]# vim /etc/httpd/conf/httpd.conf
Listen 80
[root@svr7 ~]# systemctl restart httpd
[root@svr7 ~]# ss -antlp | grep 80           #监听的端口为80
```

二、常用配置解析

P1 CentOS7 中的 web 服务

DocumentRoot: 网页文件根目录(/var/www/html)

网络路径与实际路径

网络路径: 浏览器中输入的路径 (192.168.4.7)

实际路径: 服务器上网页文件存放的路径 (/var/www/myweb)

客户端: firefox 192.168.4.7:80----》服务端 httpd ----》 /etc/httpd/conf/httpd.conf ----》
DocumentRoot /var/www/myweb----》 /var/www/myweb/index.html

网络路径: firefox 192.168.4.7/abc

实际路径: /var/www/myweb/abc

虚拟机 **svr7**:

```
[root@svr7 ~]# mkdir /var/www/myweb/abc
[root@svr7 ~]# echo wo shi abc > /var/www/myweb/abc/index.html
```

pc207 测试:

```
[root@pc207 ~]# curl 192.168.4.7/abc/
wo shi abc
```

更改配置文件 **DocumentRoot** 字段

```
[root@svr7 ~]# vim /etc/httpd/conf/httpd.conf
DocumentRoot "/webapp"
[root@svr7 ~]# mkdir /webapp
[root@svr7 ~]# echo wo shi webapp > /webapp/index.html
[root@svr7 ~]# systemctl restart httpd
```

pc207 测试

```
[root@pc207 ~]# firefox 192.168.4.7 #会发现是http的测试页面，里面没有我们写的
具体内容，这是因为有客户机地址的限制
```

P2 客户机地址限制

- 使用 **<Directory>** 配置区段

- 每个文件夹自动继承其父目录的 ACL 访问权限
- 除非针对子目录有明确设置

<Directory 目录的绝对路径>

...

Require all denied|granted

Require ip IP 或网段地址 ...

Require all denied **#拒绝所有人访问**

Require all granted **#允许所有人访问**

Require ip 192.168.4.7 **#仅 192.168.4.7 进行访问**

```
</Directory>
```

如针对/opt 目录拒绝所有人

```
<Directory /opt>
```

```
Require all denied
```

```
</Directory>
```

针对/opt/web 目录允许所有人访问

```
<Directory /opt/web>
```

```
Require all granted
```

```
</Directory>
```

注：若/opt 下面还有 nsd 目录，没有给做任何权限设置，自动继承其父目录的访问权限

若想要使用 webapp 可以成功访问，需要以下操作

```
[root@svr7 ~]# vim /etc/httpd/conf/httpd.conf
<Directory "/webapp">                                #新添加
    Require all granted                                #新添加
</Directory>                                           #新添加
[root@svr7 ~]# systemctl restart httpd
pc207测试，成功
[root@pc207 ~]# curl 192.168.4.7
```

三、虚拟 web 主机

P1 虚拟主机的含义及类型

虚拟 web 主机：由同一台服务器提供多个不同的 web 站点

区分方式：基于域名，端口，IP 地址的虚拟主机

P2 配置一个虚拟站点

• 配置文件路径

- /etc/httpd/conf/httpd.conf #主配置文件
- /etc/httpd/conf.d/*.conf #调用配置文件

为每个虚拟站点添加配置

<VirtualHost IP 地址:端口>

 ServerName 此站点的 DNS 名称

 DocumentRoot 此站点的网页根目录

</VirtualHost>

虚拟机 **svr7** 作为服务端操作

```
[root@svr7 ~]# vim /etc/httpd/conf.d/nsd01.conf
<VirtualHost *:80>                                #开启虚拟主机，在所有IP地址上启用80端口
    ServerName www.qq.com                        #指定访问的域名
    DocumentRoot /var/www/qq                     #指定网页文件路径
</VirtualHost>

<VirtualHost *:80>
    ServerName www.baidu.com
    DocumentRoot /var/www/baidu
</VirtualHost>
[root@svr7 ~]# mkdir /var/www/qq
[root@svr7 ~]# mkdir /var/www/baidu
[root@svr7 ~]# echo wo shi qq > /var/www/qq/index.html
[root@svr7 ~]# echo wo shi baidu > /var/www/baidu/index.html
[root@svr7 ~]# systemctl restart httpd
```

pc207 测试:

```
[root@pc207 ~]# vim /etc/hosts
192.168.4.7 www.qq.com www.baidu.com
[root@pc207 ~]# curl www.qq.com
wo shi qq
[root@pc207 ~]# curl www.baidu.com
wo shi baidu
```


1. 普通 NFS 服务:

1.1 NFS 共享概述

- Network File System,网络文件系统
- 用途:为客户机提供共享使用的文件夹
- 协议:NFS(TCP/UDP 2049)、RPC(TCP/UDP 111)
- 所需软件包:nfs-utils
- 系统服务:nfs-server

1.2 实验环境准备

虚拟机 svr7, 虚拟机 pc207 的防火墙默认区域设置为 trusted 或者关闭, 关闭 SELinux

```
[root@svr7 ~]# firewall-cmd --set-default-zone=trusted
[root@svr7 ~]# setenforce 0
[root@pc207 ~]# firewall-cmd --set-default-zone=trusted
[root@pc207 ~]# setenforce 0
```

svr7 和 pc207 的 yum 仓库可以正常使用

```
[root@svr7 ~]# yum clean all
[root@svr7 ~]# yum repolist
```

1.3 NFS 服务端

虚拟机 svr7 操作:

1、安装服务端软件包 nfs-utils

```
[root@svr7 ~]# yum -y install nfs-utils
```

2、创建共享目录及完成共享配置

```
[root@svr7 ~]# mkdir /test
[root@svr7 ~]# echo haha > /test/1.txt
[root@svr7 ~]# man exports    #查看帮助, 搜索example
```

```
[root@svr7 ~]# vim /etc/exports
/test 192.168.4.0/24(ro)      #文件夹路径 客户机地址(权限)
```

3、重启 nfs-server 服务

```
[root@svr7 ~]# systemctl restart nfs-server
[root@svr7 ~]# systemctl enable nfs-server
```

1.4 NFS 客户端

访问 NFS 共享

虚拟机 pc207 操作：

1、安装客户端软件：nfs-utils

```
[root@pc207 ~]# yum -y install nfs-utils
[root@pc207 ~]# rpm -q nfs-utils
```

2、挂载使用

```
[root@pc207 ~]# mkdir /nfs
[root@pc207 ~]# mount 192.168.4.7:/test /nfs
[root@pc207 ~]# df -h
[root@pc207 ~]# ls /nfs
```

3、实现开机自动挂载

_netdev：声明网络设备，系统在网络服务配置完成后,再挂载本设备

```
[root@pc207 ~]# vim /etc/fstab
192.168.4.7:/test /nfs nfs defaults,_netdev 0 0
[root@pc207 ~]# umount /nfs
[root@pc207 ~]# df -h
[root@pc207 ~]# mount -a
[root@pc207 ~]# df -h
```

1.5 案例练习

在虚拟机 svr7 上配置 NFS 共享，完成如下操作

- 1) 只读的方式访问目录/abc，只能被 192.168.4.0 系统访问
- 2) 读写的方式访问目录/public，只能被 192.168.4.0 系统访问

在虚拟机 pc207 上访问 NFS 共享目录

- 1) 将虚拟机 svr7 的/abc 挂载到本地/nfsmount
- 2) 将虚拟机 svr7 的/public 挂载到本地/nsd
- 3) 这些文件系统在系统启动时自动挂载

1. 触发挂载：

1.1 触发挂载概述

- 由 autofs 服务提供的“按需访问”机制
- 只要访问挂载点，就会触发响应，自动挂载指定设备
- 闲置超过时限（默认 5 分钟）后，会自动卸载
- 所需软件包: autofs
- 系统服务: autofs

1.2 实验环境准备

虚拟机 svr7，虚拟机 pc207 的防火墙默认区域设置为 trusted 或者关闭，关闭 SELinux

```
[root@svr7 ~]# firewall-cmd --set-default-zone=trusted
[root@svr7 ~]# setenforce 0
[root@pc207 ~]# firewall-cmd --set-default-zone=trusted
[root@pc207 ~]# setenforce 0
```

svr7 和 pc207 的 yum 仓库可以正常使用

```
[root@svr7 ~]# yum clean all
[root@svr7 ~]# yum repolist
```

1.3 autofs 配置

虚拟机 pc207 操作：

1、安装服务端软件包 autofs

```
[root@pc207 ~]# yum -y install autofs
```

2、关于分区的触发挂载

```
[root@pc207 ~]# systemctl restart autofs
[root@pc207 ~]# ls /                               #会出现misc的目录
[root@pc207 ~]# ls /misc/                          #查看里面没有数据
```

```
[root@pc207 ~]# ls -A /misc/
[root@pc207 ~]# cd /misc/aa          #失败
[root@pc207 ~]# cd /misc/bb         #失败
[root@pc207 ~]# cd /misc/cd         #成功
[root@pc207 cd]# ls
[root@pc207 cd]# df -ah             #可以查看，这是触发挂载的机制
```

1.4 autofs 配置解析

主配置文件 /etc/auto.master

--监控点目录 挂载配置文件的路径

挂载配置文件，比如 /etc/auto.misc

--触发点目录 -挂载参数 :设备名

查看这两个文件即可，不做更改

```
[root@pc207 ~]# vim /etc/auto.master
[root@pc207 ~]# vim /etc/auto.misc
```

1.5 案例练习触发挂载一个分区

```
[root@pc207 ~]# fdisk /dev/sdb      #划分一个主分区，大小为3G
```

```
[root@pc207 ~]# mkfs.xfs /dev/sdb1
```

```
[root@pc207 ~]# blkid /dev/sdb1
```

当触发/misc/mydev时，实现自动挂载

```
[root@pc207 ~]# ls /misc/mydev
```

ls: 无法访问/misc/mydev: 没有那个文件或目录

创建触发式挂载

```
[root@pc207 ~]# vim /etc/auto.misc
```

```
mydev -fstype=xfs  :/dev/sdb1
```

```
[root@pc207 ~]# systemctl restart autofs
```

```
[root@pc207 ~]# ls /misc/mydev  
[root@pc207 ~]# df -ah
```

#可以查看，没有报错

1.6 触发挂载进阶

实现 NFS 的触发挂载

虚拟机 svr7 为服务端，构建 NFS 共享

1、安装软件包 nfs-utils

```
[root@svr7 ~]# rpm -q nfs-utils
```

2、创建共享目录及完成共享配置

```
[root@svr7 ~]# mkdir /test  
[root@svr7 ~]# echo haha > /test/1.txt  
[root@svr7 ~]# vim /etc/exports  
/test 192.168.4.0/24(ro)
```

3、重启 nfs-server 服务

```
[root@svr7 ~]# systemctl restart nfs-server  
[root@svr7 ~]# systemctl enable nfs-server
```

虚拟机 pc207 作为客户端测试访问

1、查看服务端共享

```
[root@pc207 ~]# showmount -e 192.168.4.7
```

2、以/misc/目录创建触发式挂载（仅修改/etc/auto.misc 指定触发点目录）

```
[root@pc207 ~]# ls /misc  
[root@pc207 ~]# cd /misc/autonfs  
ls: 无法访问/misc/autonfs: 没有那个文件或目录  
[root@pc207 ~]# vim /etc/auto.misc  
autonfs -fstype=nfs 192.168.4.7:/test
```

```
[root@pc207 ~]# systemctl restart autofs
[root@pc207 ~]# ls /misc
[root@pc207 ~]# cd /misc/autofs
[root@pc207 autofs]# df -ah
```

2. DNS 服务概述:

2.1 DNS 解析的作用

DNS 服务器的功能

- 正向解析:根据注册的域名查找其对应的 IP 地址
- 反向解析:根据 IP 地址查找对应的注册域名,不常用

一级 DNS 服务器:专门负责一级域名的解析

二级 DNS 服务器:专门负责二级域名的解析

三级 DNS 服务器:专门负责三级域名的解析

根域名 DNS 服务器:专门负责根域名的解析

2.2 DNS 的分布式结构

域名体系: 所有的域名必须以点结尾

www.qq.com. www.baidu.com.

根域名: (.)

一级域名: .cn .us .tw .kr .jp .hk

二级域名: .com.cn .edu.cn .org.cn .net.cn

三级域名: sina.com.cn .nb.com.cn .haxi.com.cn

2.3 BIND 域名服务

完全合格的主机名(Full Qualified Domain Name)FQDN:

站点名+注册申请的域名后缀

- BIND 服务器端程序
 - 主要执行程序: /usr/sbin/named
 - 系统服务: named
 - 默认端口: TCP/UDP 53
 - 运行时的虚拟根环境: /var/named/chroot/
- 主配置文件: **/etc/named.conf** #主要 设置负责解析的域名 tedu.cn
- 地址库文件: /var/named/ #域名解析的结果, 完整的主机名与 IP 地址对应关系

2.4 构建单区域 DNS

服务端虚拟机 svr7 操作:

1、安装软件包 bind、bind-chroot

```
[root@svr7 ~]# yum -y install bind-chroot bind
```

#bind是域名服务包, bind-chroot 是提供虚拟根支持

2、修改主配置/etc/named.conf

```
[root@svr7 ~]# cp /etc/named.conf /etc/named.bak
```

```
[root@svr7 ~]# vim /etc/named.conf
```

```
options {
```

```
    directory "/var/named";                      #指定地址库文件位置
```

```
};
```

```
zone "tedu.cn" IN {                      #指定本机负责解析的域名
```

```
    type master;                      #指定本机为主DNS服务器
```



```
file "tedu.cn.zone";    #指定地址库文件名称（在/var/named目录下创建的地址库文件要和这个一致）  
};
```

3、建立地址库文件 `tedu.cn.zone`

-p:保持权限不变进行复制

原理:让 `named` 用户对地址库文件有权限

```
[root@svr7 ~]# cd /var/named/  
[root@svr7 named]# cp -p named.localhost tedu.cn.zone  
[root@svr7 named]# vim tedu.cn.zone  
  
#所有的域名都要写 点 作为结尾  
#没有点作为结尾，默认补全本地址库负责的域名  
  
$TTL 1D  
@ IN SOA @ rname.invalid. (  
                                0 ; serial  
                                1D ; refresh  
                                1H ; retry  
                                1W ; expire  
                                3H ) ; minimum  
  
tedu.cn. NS svr7.tedu.cn.    #声明维护tedu.cn.域名服务器叫什么名称  
svr7.tedu.cn.  A  192.168.4.7    #指定DNS服务IP地址  
www.tedu.cn.  A  1.2.3.4  
ftp.tedu.cn.  A  5.6.7.8
```

4、重起 `named` 服务

```
[root@svr7 named]# systemctl restart named
```

客户端 `pc207` 操作:

指定 `DNS` 服务器地址，并验证

```
[root@pc207 ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf
```

```
[root@pc207 ~]# nslookup www.tedu.cn
```

```
Server: 192.168.4.7
```

```
Address: 192.168.4.7#53
```

```
Name: www.tedu.cn
```

```
Address: 1.2.3.4
```

1. DNS 服务概述:

1.1 构建单区域 DNS

服务端虚拟机 svr7 操作:

1、安装软件包 bind、bind-chroot

```
[root@svr7 ~]# yum -y install bind-chroot bind
```

#bind是域名服务包, bind-chroot 是提供虚拟根支持

2、修改主配置/etc/named.conf

```
[root@svr7 ~]# cp /etc/named.conf /etc/named.bak
```

```
[root@svr7 ~]# vim /etc/named.conf
```

```
options {
```

```
    directory "/var/named";      #指定地址库文件位置
```

```
};
```

```
zone "tedu.cn" IN {              #指定本机负责解析的域名
```

```
    type master;                 #指定本机为主DNS服务器
```

```
    file "tedu.cn.zone";         #指定地址库文件名称 (在/var/named目录下创建的地址  
库文件要和这个一致)
```

```
};
```

3、建立地址库文件 tedu.cn.zone

-p:保持权限不变进行复制

原理:让 named 用户对地址库文件有权限

```
[root@svr7 ~]# cd /var/named/
```

```
[root@svr7 named]# cp -p named.localhost tedu.cn.zone
```

```
[root@svr7 named]# vim tedu.cn.zone
```

#所有的域名都要写 点 作为结尾

#没有点作为结尾，默认补全本地址库负责的域名

\$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H) ; minimum

tedu.cn. NS svr7.tedu.cn. #声明维护tedu.cn.域名服务器叫什么名称

svr7.tedu.cn. A 192.168.4.7 #指定DNS服务IP地址

www.tedu.cn. A 1.2.3.4

ftp.tedu.cn. A 5.6.7.8

4、重起 named 服务

```
[root@svr7 named]# systemctl restart named
```

客户端 pc207 操作：

指定 DNS 服务器地址，并验证

```
[root@pc207 ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf
```

```
[root@pc207 ~]# yum -y install bind-utils
```

```
[root@pc207 ~]# nslookup www.tedu.cn
```

Server: 192.168.4.7

Address: 192.168.4.7#53

Name: www.tedu.cn

Address: 1.2.3.4

1.2 构建多区域 DNS

搭建多区域的 DNS 服务器，即负责多个域名的解析

在虚拟机 svr7 实现 DNS 服务器构建，负责解析 baidu.com 域名 最终实现客户端解析
www.baidu.com----->10.20.30.40

服务端虚拟机 svr7 操作：

1、修改主配置文件/etc/named.conf，添加 zone 配置

```
[root@svr7 /]# vim /etc/named.conf
```

#主配置文件下面新添加如下部分

```
zone "baidu.com" IN {  
type master;  
file "baidu.com.zone";  
};
```

3、建立地址库文件 baidu.com.zone

```
[root@svr7 ~]# cd /var/named/
```

```
[root@svr7 named]# cp -p tedu.cn.zone baidu.com.zone
```

```
[root@svr7 named]# vim baidu.com.zone
```

```
$TTL 1D
```

```
@ IN SOA @ rname.invalid. (
```

```
0 ; serial
```

```
1D ; refresh
```

```
1H ; retry
```

```
1W ; expire
```

```
3H ) ; minimum
```

```
baidu.com. NS svr7
```

```
svr7 A 192.168.4.7
```

```
www A 10.20.30.40
```

```
ftp A 50.60.70.80
```

4、重起 named 服务

```
[root@svr7 named]# systemctl restart named
```

客户端 pc207 操作:

指定 DNS 服务器地址, 并验证

```
[root@pc207 ~]# nslookup www.tedu.cn
```

```
[root@pc207 ~]# nslookup www.baidu.com
```

2. 特殊的解析记录:

2.1 基于解析记录的轮询 (负载均衡, 缓解网站服务器的压力)

基于 DNS 的站点负载均衡

一个域名——》多个不同 IP 地址

每个 IP 提供镜像服务内容

虚拟机 svr7:

```
[root@svr7 named]# vim baidu.com.zone
```

```
$TTL 1D
```

```
@ IN SOA @ rname.invalid. (
```

```
0 ; serial
```

```
1D ; refresh
```

```
1H ; retry
```

```
1W ; expire
```

```
3H ) ; minimum
```

```
baidu.com. NS svr7
```

```
svr7 A 192.168.4.7
```

```
www A 192.168.4.50
```

```
www A 192.168.4.60
```

```
www A 192.168.4.70
ftp A 50.60.70.80
[root@svr7 /]# systemctl restart named
```

虚拟机 pc207 验证:

```
[root@pc207 ~]# ping www.baidu.com #使用ping测试, 对应ip地址发生改变
```

2.2 泛域名解析

作用: 解决用户输入域名时讲站点输入错误的问题

虚拟机 svr7:

```
[root@svr7 named]# vim baidu.com.zone
.....
* A 7.8.9.10
[root@svr7 ~]# systemctl restart named
```

虚拟机 pc207 验证

```
[root@pc207 ~]# nslookup www.baidu.com           #成功
[root@pc207 ~]# nslookup xixi.baidu.com           #成功
[root@pc207 ~]# nslookup baidu.com                #失败
```

需要在svr7主机的配置文件里面添加以下信息

```
[root@svr7 ~]# vim /var/named/baidu.com.zone
.....
baidu.com. A 70.80.90.100
[root@svr7 ~]# systemctl restart named
```

pc207测试:

```
[root@pc207 ~]# nslookup baidu.com #成功
```

2.3 有规律的泛域名解析

pc1.baidu.com----->192.168.10.1

pc2.baidu.com----->192.168.10.2

pc3.baidu.com----->192.168.10.3

.....

pc100.baidu.com----->192.168.10.100

内置函数: \$GENERATE 造数

\$GENERATE 1-50 pc\$ A 192.168.10.\$

虚拟机 svr7:

```
[root@svr7 /]# vim /var/named/baidu.com.zone
```

...

```
$GENERATE 1-100 pc$ A 192.168.10.$
```

```
[root@svr7 ~]# systemctl restart named
```

虚拟机 pc207:

```
[root@pc207 ~]# nslookup pc1.baidu.com #会出对应的IP
```

```
[root@pc207 ~]# nslookup pc2.baidu.com
```

```
[root@pc207 ~]# nslookup pc100.baidu.com
```

```
[root@pc207 ~]# nslookup pc101.baidu.com #出现匹配的IP为7.8.9.10, 因为配置文件有  
泛域名解析
```

2.4 解析记录的别名 CNAME

虚拟机 svr7:

```
[root@svr7 /]# vim /var/named/baidu.com.zone
```

.....

```
tts CNAME ftp
```

```
[root@svr7 ~]# systemctl restart named
```

pc207 测试


```
[root@pc207 ~]# nslookup tts.baidu.com
```

3. 主从 DNS 架构

3.1 什么是主/从 DNS 服务器

主从 DNS 架构

DNS 服务器的主从结构，备份主服务器的数据，解决单点故障

3.2 主从 DNS 服务器环境准备

- 1.虚拟机 svr7：主 DNS 服务器 192.168.4.7
- 2.虚拟机 pc207：从 DNS 服务器 192.168.4.207
- 3.虚拟机 C：客户端测试 192.168.4.10

虚拟机 svr7 构建主 DNS 服务器：

1.修改主配置文件,指定从服务器 IP 地址

```
[root@svr7 /]# vim /etc/named.conf

options {
    directory "/var/named";
    allow-transfer { 192.168.4.207; };    #指定从服务器
};

...
```

2.修改地址库文件，进行声明从服务器

```
[root@svr7 /]# vim /var/named/tedu.cn.zone

.....

tedu.cn. NS svr7
tedu.cn. NS pc207    #声明从服务器
svr7 A 192.168.4.7
```

```
pc207 A 192.168.4.207
www A 1.2.3.4
[root@svr7 ~]# systemctl restart named
```

3.3 部署从 DNS 服务器

虚拟机 pc207 构建从服务器：

1. 安装软件包

```
[root@pc207 ~]# yum -y install bind bind-chroot
```

2. 修改主配置文件

```
[root@pc207 ~]# vim /etc/named.conf
options {
    directory "/var/named";
};
zone "tedu.cn" IN {
    type slave;
    file    "/var/named/slaves/tedu.cn.slave" ;
    masters { 192.168.4.7; };           #指定主DNS服务器位置
};
[root@pc207 ~]# systemctl restart named
```

虚拟机 C 测试

```
[root@C opt]# nslookup www.tedu.cn 192.168.4.7           #指定DNS测试
[root@C opt]# nslookup www.tedu.cn 192.168.4.207
[root@C opt]# vim /etc/resolv.conf
nameserver 192.168.4.7
nameserver 192.168.4.207
[root@C opt]# nslookup www.tedu.cn
```

3.4 模拟主 DNS 宕机

```
[root@svr7 ~]# systemctl stop named
```

虚拟机 C 测试，还可以访问，此时是从 DNS 提供的解析

```
[root@C opt]# nslookup www.tedu.cn
```

3.5 主从 DNS 服务器同步数据

- 1.修改数据必定是在主 DNS 服务器上完成
- 2.必须修改数据版本号,必须往大的方向修改

地址库文件内容:

2019122401 ; serial #数据版本号由 10 个数字组成

1D ; refresh #每隔 1 天,主从会进行同步数据

1H ; retry #重试时间间隔,每隔 1 个小时

1W ; expire #失效时间,1 个星期

3H) ; minimum #无效记录的缓存时间,3 个小时

主DNS服务器修改地址库文件:

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
..... www A 1.2.3.5
```

```
[root@svr7 ~]# systemctl restart named
```

虚拟机C测试

```
[root@C opt]# nslookup www.tedu.cn 192.168.4.7 #显示解析地址为1.2.3.5
```

```
[root@C opt]# nslookup www.tedu.cn 192.168.4.207 #显示解析地址为1.2.3.4，没有同步
```

同步数据测试 (需要修改主服务器地址库文件数据版本号重启服务才能同步数据)

```
[root@svr7 ~]# vim /var/named/tedu.cn.zone
```

```
$TTL 1D
```

```
@ IN SOA @ rname.invalid. (
```

```
2020010101 ; serial #需要修改数据版本号（只能在原基础往大修改，例如现在是  
2020010101，若修改了地址库数据，需要同步则需要修改的数必须大于2020010101）
```

```
.....
```

```
[root@svr7 ~]# systemctl restart named
```

虚拟机C测试：

```
[root@C opt]# nslookup www.tedu.cn 192.168.4.207      #同步成功
```

1. 电子邮件通信:

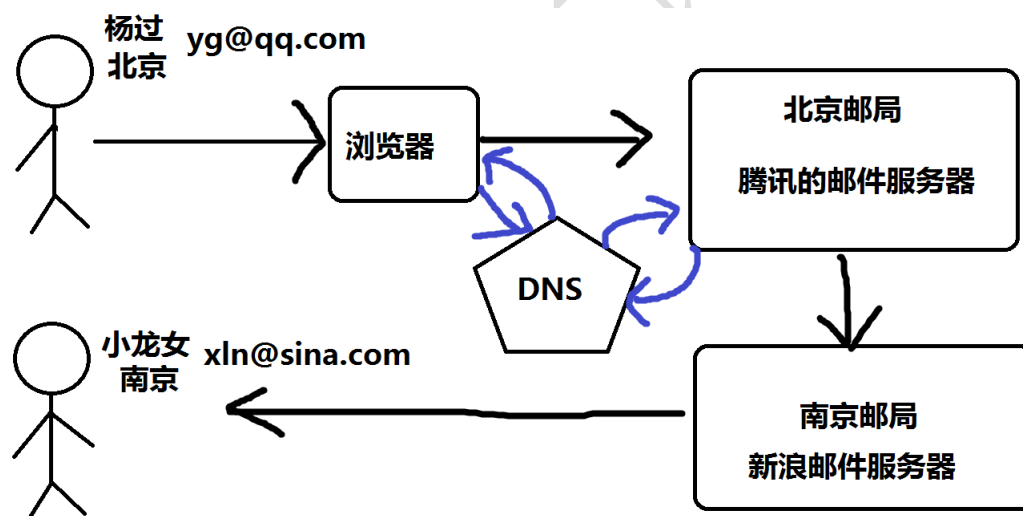
1.1 基础邮件服务

SMTP: 用户发邮件协议(25)

POP3: 用户收邮件协议(110)

电子邮件通信

- 电子邮件服务器的基本功能
 - 为用户提供电子邮箱存储空间(用户名@邮件域名)
 - 处理用户发出的邮件 —— 传递给收件服务器
 - 处理用户收到的邮件 —— 投递到邮箱



服务端虚拟机 svr7 操作:

1、安装软件包 bind、bind-chroot

```
[root@svr7 ~]# yum -y install bind-chroot bind
```

2、修改主配置/etc/named.conf

```
[root@svr7 ~]# cp /etc/named.conf /etc/named.bak
```

```
[root@svr7 ~]# vim /etc/named.conf
options {
directory "/var/named";      #指定地址库文件位置
};
zone "example.com" IN {
    type master;
    file "example.com.zone";
};
```

3、建立地址库文件 `tedu.cn.zone`

```
[root@svr7 ~]# cd /var/named/
[root@svr7 named]# cp -p named.localhost example.com.zone
[root@svr7 named]# vim example.com.zone
$TTL 1D
@ IN SOA @ rname.invalid. (
                                0 ; serial
                                1D ; refresh
                                1H ; retry
                                1W ; expire
                                3H ) ; minimum

example.com.  NS   svr7
example.com.  MX   10  mail      #MX邮件交换记录，10为第几台邮件服务器，
                                #数字越小优先级越高，mail，邮件交换（收发）主要由mail负责
svr7          A    192.168.4.7
mail          A    192.168.4.207 #指定mail是谁
```

4、重起 `named` 服务

```
[root@svr7 named]# systemctl restart named
```

客户端 pc207 操作:

Pc207 主机验证邮件交换记录

```
[root@pc207 ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf
[root@pc207 ~]# yum -y install bind-utils
[root@pc207 ~]# host -t MX example.com    #查看在example.com域中邮件服务器是谁
[root@pc207 ~]# host mail.example.com    #查看mail.example.com的IP是谁
```

1.2 mail 命令发信/收信

1.2.1 快速部署 postfix 邮件服务器

1.检查是否安装postfix软件包

```
[root@pc207 named]# rpm -q postfix
```

2.修改配置文件

```
[root@pc207 named]# vim /etc/postfix/main.cf
```

```
99 myorigin = example.com    #默认补全的域名后缀
```

```
116 inet_interfaces = all    #监听接口，本机所有网卡都提供邮件功能
```

```
164 mydestination = example.com    #判定为本域邮件的依据
```

3.重启服务

```
[root@pc207 named]# systemctl restart postfix
```

1.2.2 使用 mail 命令发信/收信

创建本地用户yg和xln用于测试

```
[root@pc207 ~]# useradd yg
```

```
[root@pc207 ~]# useradd xln
```

• mail 发信操作: mail -s '邮件标题' -r 发件人 收件人

```
[root@pc207 ~]# mail -s 'test01' -r yg xln
```

```
hahaxixi    #邮件内容
```

```
.    #输入点结束，退出
```

EOT

- mail 收信操作: mail [-u 用户名]

```
[root@pc207 ~]# mail -u xln
```

Heirloom Mail version 12.5 7/5/10. Type ? for help.

"/var/mail/yg": 1 message 1 new

>N 1 yg@example.com Fri Jul 24 13:17 18/548 "test01"

& 1 #输入编号1

Message 1:

...

& exit #退出

非交互式发送邮件:

echo 邮件内容 | mail -s '邮件标题' -r 发件人 收件人

```
[root@pc207 ~]# echo 123456 zhangsan | mail -s 'test02' -r yg xln
```

```
[root@pc207 ~]# mail -u xln
```

2. DNS 分离解析:

2.1 分离解析概述

当收到客户机的 DNS 查询请求的时候

- 能够区分客户机的来源地址
- 为不同类别的客户机提供不同的解析结果(IP 地址)
- 不同类别的客户机, 在解析相同的一个域名, 得到的解析结果不同

----->A 类客户端-----》192.168.1.1 www.qq.com -

----->B 类客户端-----》10.11.12.13

----->网通的客户端-----》网通机房的服务器地址 www.qq.com

----->电信的客户端-----》电信机房的服务器地址

– 为客户端提供最近的资源服务器

BIND 的 view 视图

• 根据源地址集合将客户机分类

–不同客户机获得不同结果(待遇有差别)

– view 分类，客户端匹配由上及下进行匹配，匹配及停止

– view 分类要合理，所有客户端都必须找到自己的类别

– 所有的 zone 都必须在 view 大括号中

2.2 配置分离解析

环境及需求

– 权威 DNS: svr7.tedu.cn 192.168.4.7

– 负责区域: tedu.cn

– A 记录分离解析 —— 以 www.tedu.cn 为例

客户机来自	解析结果
192.168.4.207	192.168.4.100
其他地址	1.2.3.4

```
[root@svr7 /]# vim /etc/named.conf
```

```
.....
```

```
view "NSD" {                                     #设置分类名称
    match-clients { 192.168.4.207; };           #匹配客户端来源
    zone "tedu.cn" IN {
        type master;
        file "tedu.cn.zone";                   #解析结果192.168.4.100
    };
}
```

```
};  
};  
  
view "other" {                                #设置分类名称  
    match-clients { any; };                  #匹配客户端来源  
    zone "tedu.cn" IN {  
        type master;  
        file "tedu.cn.other";    #解析结果1.2.3.4  
    };  
};  
  
[root@svr7 /]# vim /var/named/tedu.cn.zone  
.....  
www      A      192.168.4.100  
  
[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.other  
[root@svr7 named]# vim tedu.cn.other  
.....  
www      A      1.2.3.4  
  
[root@svr7 /]# systemctl restart named  
分别用客户端pc207, 虚拟机A验证  
[root@pc207 ~]# nslookup www.tedu.cn  
[root@C opt]# nslookup www.tedu.cn
```

2.3 多区域分离解析

分类(匹配客户端来源相同)相同:

客户机来自	域名	解析结果
192.168.4.207	www.tedu.cn www.qq.com	192.168.4.100
其他地址	www.tedu.cn www.qq.com	1.2.3.4

1.修改主配置/etc/named.conf

```
[root@svr7 /]# vim /etc/named.conf

.....
view "nsd" {                                #设置分类名称
    match-clients { 192.168.4.207; };      #匹配客户端来源
    zone "tedu.cn" IN {
        type master;
        file "tedu.cn.zone";  ----》解析结果192.168.4.100
    };
    zone "qq.com" IN {
        type master;
        file "qq.com.zone";  ----》解析结果192.168.4.100
    };
};

view "other" {                               #设置分类名称
    match-clients { any; };                 #匹配客户端来源
    zone "tedu.cn" IN {
        type master;
        file "tedu.cn.other";  ----》 解析结果1.2.3.4
    };
    zone "qq.com" IN {
```

```
type master;  
file "qq.com.other"; ----》 解析结果1.2.3.4  
};  
};
```

2.建立地址库文件

```
[root@svr7 ~]# cd /var/named/  
[root@svr7 named]# cp -p tedu.cn.zone qq.com.zone  
[root@svr7 named]# cp -p tedu.cn.zone qq.com.other  
[root@svr7 named]# vim qq.com.zone  
.....  
qq.com.      NS      svr7  
svr7         A        192.168.4.7  
www          A        192.168.4.100  
  
[root@svr7 named]# vim qq.com.other  
.....  
qq.com.      NS      svr7  
svr7         A        192.168.4.7  
www          A        1.2.3.4  
  
[root@svr7 named]# systemctl restart named
```

3.验证, pc207 和虚拟机 A 验证, 结果不同

```
[root@pc207 /]# nslookup www.tedu.cn  
[root@pc207 /]# nslookup www.qq.com  
[root@C /]# nslookup www.tedu.cn  
[root@C /]# nslookup www.qq.com
```

作业:

分类(匹配客户端来源不同) 不同

如要实现以下案例要求:

客户端 192.168.4.207 www.tedu.cn 192.168.4.100

其他地址 www.tedu.cn 1.2.3.4

客户端 192.168.4.10 www.qq.com 192.168.10.100

其他地址 www.qq.com 172.25.0.11

达内Linux云计算学院

1. DNS 分离解析:

1.1 多区域分离解析

分类(匹配客户端来源不相同)不同:

如要实现以下案例要求:

客户端 192.168.4.207-----> www.tedu.cn -----> 192.168.4.100

其他地址 -----> www.tedu.cn -----> 1.2.3.4

客户端 192.168.4.10 -----> www.qq.com -----> 192.168.10.100

其他地址 -----> www.qq.com -----> 172.25.0.11

根据上述题目要求, 可以得出在客户端 192.168.4.207 访问 www.tedu.cn 返回的结果是 192.168.4.100 之外, 其他无论是在 192.168.4.207 访问 www.qq.com, 还是在其他主机访问 www.tedu.cn, 返回的都是其他的结果 得出结果如图所示:

客户机来自	域名	解析结果
192.268.4.207	www.tedu.cn	192.268.4.100
	www.qq.com	172.25.0.11
192.168.4.10	www.tedu.cn	1.2.3.4
	www.qq.com	192.168.10.100
其他地址	www.tedu.cn	1.2.3.4
	www.qq.com	172.25.0.11

1.修改主配置/etc/named.conf

```
[root@svr7 /]# vim /etc/named.conf

options {
    directory "/var/named";
};

view "nsd" {
    match-clients { 192.168.4.207; };
    zone "tedu.cn" IN {
        type master;
        file "tedu.cn.zone"; ----》解析结果192.168.4.100
    };
    zone "qq.com" IN {
        type master;
        file "qq.com.zone"; ----》解析结果172.25.0.11
    };
};

view "vip" {
    match-clients { 192.168.4.10; };
    zone "tedu.cn" IN {
        type master;
        file "tedu.cn.other"; ----》解析结果1.2.3.4
    };
    zone "qq.com" IN {
        type master;
        file "qq.com.other"; ----》解析结果192.168.10.100
    };
};

view "other" {
```

```
match-clients { any; };  
zone "tedu.cn" IN {  
    type master;  
    file "tedu.cn.other"; ----》解析结果1.2.3.4  
};  
zone "qq.com" IN {  
    type master;  
    file "qq.com.zone"; ----》解析结果172.25.0.11  
};  
};
```

2.建立地址库文件

```
[root@svr7 /]# vim /var/named/qq.com.zone  
.....  
qq.com. NS svr7  
svr7 A 192.168.4.7  
www A 172.25.0.11  
[root@svr7 /]# vim /var/named/qq.com.other  
.....  
qq.com. NS svr7  
svr7 A 192.168.4.7  
www A 192.168.10.100  
[root@svr7 named]# systemctl restart named
```

验证: pc207 主机和虚拟机 A 验证

```
[root@pc207 /]# nslookup www.tedu.cn  
[root@pc207 /]# nslookup www.qq.com  
[root@C /]# nslookup www.tedu.cn  
[root@C /]# nslookup www.qq.com
```


2. 缓存 DNS 服务:

2.1 缓存 DNS 作用及适用场景

作用: 缓存解析记录, 加快解析

适用场景:

--互联网出口带宽较低的企业局域网络

--ISP 服务商的公共 DNS 服务器

2.2 构建缓存 DNS 服务

虚拟机 svr7: 作为真实的 DNS 服务器

虚拟机 pc207: 作为缓存 DNS 服务器

虚拟机 A: 作为客户端

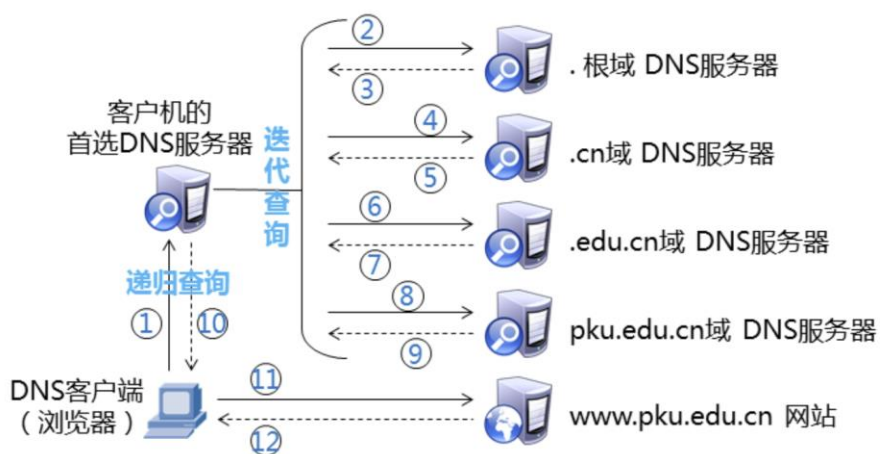
虚拟机 pc207:构建缓存 DNS 服务器

```
[root@pc207 /]# vim /etc/named.conf
options {
    directory "/var/named";
    forwarders { 192.168.4.7; }; #请求转发给192.168.4.7
};
[root@pc207 /]# systemctl restart named
```

虚拟机 A 进行测试:

```
[root@C ~]# nslookup www.tedu.cn 192.168.4.207
```

2.3 DNS 查询方式



递归解析: 客户端的解析请求,到主 DNS 服务器,然后主 DNS 服务器与其他 DNS 服务沟通,最终将解析结果带回来的过程

迭代解析: 主 DNS 服务器指引下一个 DNS 服务器地址

关闭递归解析:

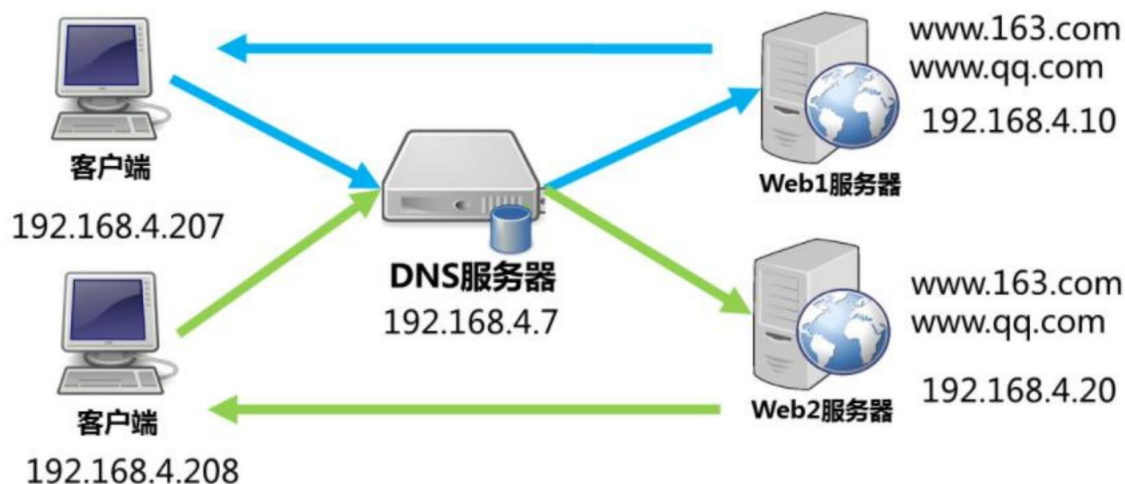
```
[root@svr7 ~]# vim /etc/named.conf

options {
    directory "/var/named";
    recursion no; #关闭递归解析
};
```

3. WEB 服务项目实战:

3.1 案例需求要点:

web 服务器和 DNS 服务结合 (web 服务器做需要开启基于域名的虚拟主机, DNS 需要使用分离解析技术)



	主机名	IP 地址
虚拟机 A	A.tedu.cn	192.168.4.10
虚拟机 B	B.tedu.cn	192.168.4.20
虚拟机 C	C.tedu.cn	192.168.4.208
虚拟机 svr7	svr7.tedu.cn	192.168.4.7
虚拟机 pc207	pc207.tedu.cn	192.168.4.207

3.2 实践步骤:

虚拟机 A 操作:

1. 安装软件包 httpd

```
[root@A ~]# yum -y install httpd
```

2. 建立修改调用配置文件

```
[root@A ~]# vim /etc/httpd/conf.d/nsd01.conf
<VirtualHost *:80>
    ServerName www.qq.com
```

```
DocumentRoot /var/www/qq
</VirtualHost>
<VirtualHost *:80>
    ServerName www.163.com
    DocumentRoot /var/www/163
</VirtualHost>
[root@A ~]# mkdir /var/www/qq /var/www/163
[root@A ~]# echo '<h1> Web1 QQ' > /var/www/qq/index.html
[root@A ~]# echo '<h1> Web1 163' > /var/www/163/index.html
[root@A ~]# systemctl restart httpd
```

虚拟机 B 操作:

```
[root@D ~]# yum -y install httpd
从虚拟机A拷贝nsd01配置文件
[root@A ~]# scp /etc/httpd/conf.d/nsd01.conf 192.168.4.20:/etc/httpd/conf.d/
[root@B ~]# mkdir /var/www/qq /var/www/163
[root@B ~]# echo '<h1>Web2 QQ' > /var/www/qq/index.html
[root@B ~]# echo '<h1>Web2 163' > /var/www/163/index.html
[root@B ~]# systemctl restart httpd
```

虚拟机 svr7 操作:

1、修改主配置文件

```
[root@svr7 ~]# vim /etc/named.conf
.....
view "vip" {
    match-clients { 192.168.4.207; };
    zone "163.com" IN {
        type master;
        file "163.com.zone";
```

```
};  
zone "qq.com" IN {  
    type master;  
    file "qq.com.zone";  
};  
};  
view "other" {  
    match-clients { any; };  
    zone "163.com" IN {  
        type master;  
        file "163.com.other";  
    };  
    zone "qq.com" IN {  
        type master;  
        file "qq.com.other";  
    };  
};
```

2、建立地址库文件

```
[root@svr7 ~]# cd /var/named/  
[root@svr7 named]# cp -p qq.com.zone 163.com.zone  
[root@svr7 named]# cp -p qq.com.zone 163.com.other  
[root@svr7 named]# vim 163.com.zone  
.....  
163.com. NS svr7  
svr7 A 192.168.4.7  
www A 192.168.4.10  
[root@svr7 named]# vim qq.com.zone  
.....
```

```
qq.com. NS svr7
svr7 A 192.168.4.7
www A 192.18.4.10
[root@svr7 named]# vim 163.com.other
...
163.com. NS svr7
svr7 A 192.168.4.7
www A 192.168.4.20
[root@svr7 named]# vim qq.com.other
.....
qq.com. NS svr7
svr7 A 192.168.4.7
www A 192.168.4.20
[root@svr7 named]# systemctl restart named
```

测试：指定 DNS 服务器地址

虚拟机pc207

```
[root@pc207 ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf
[root@pc207 ~]# curl www.qq.com
<h1>web1 qq
[root@pc207 ~]# curl www.163.com
<h1>web1 163
```

虚拟机C

```
[root@C ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf
[root@C ~]# curl www.qq.com
<h1>web2 qq
[root@C ~]# curl www.163.com
<h1>web2 163
```