

# NSD SERVICES DAY06

1. [案例1：rsync基本用法](#)
2. [案例2：rsync+SSH同步](#)
3. [案例3：配置rsync服务端](#)
4. [案例4：访问rsync共享资源](#)
5. [案例5：使用inotifywait工具](#)
6. [案例6：配置Web镜像同步](#)
7. [案例7：配置并验证Split分离解析](#)

## 1 案例1：rsync基本用法

### 1.1 问题

本例要求掌握远程同步的基本操作，使用rsync命令完成下列任务：

1. 将目录 /boot 同步到目录 /todir 下
2. 将目录 /boot 下的文档同步到目录 /todir 下
3. 在目录 /boot 下新增文件 a.txt，删除 /todir 下的子目录 grub2，再次同步使 /todir 与 /boot 一致
4. 验证 -a、-n、-v、--delete 选项的含义

### 1.2 方案

本地同步操作：

- rsync [选项...] 本地目录1 本地目录2
- rsync [选项...] 本地目录1/ 本地目录2

rsync同步工具的常用选项：

- -n：测试同步过程，不做实际修改
- --delete：删除目标文件夹内多余的文档
- -a：归档模式，相当于-rlptgoD
- -v：显示详细操作信息

[Top](#)

- -z：传输过程中启用压缩/解压

## 1.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：rsync同步基本操作

1) 将目录 /boot 同步到目录 /todir 下

```
01. [root@svr7 ~]# ls -l /todir //同步前
02. ls: 无法访问/todir: 没有那个文件或目录
03. [root@svr7 ~]# rsync -a /boot /todir //将目录1作为目录2的子目录
04. [root@svr7 ~]# ls -l /todir //检查同步结果
05. 总用量 4
06. dr-xr-xr-x. 4 root root 4096 11月 30 18:50 boot
```

2) 将目录 /boot 下的文档同步到目录 /todir 下

```
01. [root@svr7 ~]# rm -rf /todir //清理掉目录2
02. [root@svr7 ~]# rsync -a /boot/ /todir //将目录1下的文档同步到目录2下
03. [root@svr7 ~]# ls -l /todir //检查同步结果
04. 总用量 126708
05. -rw-r--r--. 1 root root 126426 10月 30 2015 config-3.10.0-327.el7.x86_64
06. drwxr-xr-x. 2 root root 4096 11月 30 18:50 extlinux
07. drwx-----. 6 root root 104 12月 9 09:58 grub2
08. ...
```

[Top](#)

### 3 ) 同步效果测试

在目录/boot下新增文件a.txt , 删除/todir下的子目录 grub2 :

01. [ root@svr7 ~] # touch /boot/a.txt
02. [ root@svr7 ~] # rm -rf /todir/grub2/

现在目录/boot和/todir目录下的内容已经不一致了 :

01. [ root@svr7 ~] # ls -ld /boot/a.txt /todir/a.txt
02. ls: 无法访问/todir/a.txt: 没有那个文件或目录
03. -rw-r--r--. 1 root root 0 1月 11 21:09 /boot/a.txt
04. [ root@svr7 ~] # ls -ld /boot/grub2 /todir/grub2
05. ls: 无法访问/todir/grub2: 没有那个文件或目录
06. drwx-----. 6 root root 104 12月 9 09:58 /boot/grub2

再次同步使/todir与/boot一致 :

01. [ root@svr7 ~] # rsync -a /boot/ /todir/

确认同步结果 :

01. [ root@svr7 ~] # ls -ld /boot/a.txt /todir/a.txt
02. -rw-r--r--. 1 root root 0 1月 11 21:09 /boot/a.txt

[Top](#)

```
03. -rw-r--r--. 1 root root 0 1月 11 21:09 /todir/a.txt
04. [root@svr7 ~]# ls -ld /boot/grub2 /todir/grub2
05. drwx-----. 6 root root 104 12月 9 09:58 /boot/grub2
06. drwx-----. 6 root root 104 12月 9 09:58 /todir/grub2
```

## 步骤二：验证 -a、-v、-n、--delete 选项的含义

### 1) 验证-a选项

当目录1包含文件夹时，若缺少-a或-r选项则文件夹会被忽略：

```
01. [root@svr7 ~]# rsync /home /testa
02. skipping directory home
03. [root@svr7 ~]# ls -ld /testa
04. ls: 无法访问/testa: 没有那个文件或目录
```

添加-a后才会执行同步：

```
01. [root@svr7 ~]# rsync -a /home/ /testa
02. [root@svr7 ~]# ls -ld /testa
03. drwxr-xr-x. 4 root root 31 1月 6 17:33 /testa
```

类似的情况，当目录1中的数据出现权限、归属、修改时间等变化时，若文件内容不变默认不会同步，若希望目录2也同步这些变化，也需要-a选项。

### 2) 验证-v选项

创建测试目录及文档：

[Top](#)

01. [root@svr7 ~] # mkdir /fdir
02. [root@svr7 ~] # touch /fdir/1.txt

添加-v选项时，可以看到操作细节信息，比如第一次同步时：

01. [root@svr7 ~] # rsync -av /fdir/ /tdir
02. sending incremental file list
03. created directory /tdir
04. ./
05. 1.txt //传输文档列表
- 06.
07. sent 82 bytes received 34 bytes 232.00 bytes/sec
08. total size is 0 speedup is 0.00

在目录/fdir/添加文件2.txt，再次跟踪同步信息：

01. [root@svr7 ~] # touch /fdir/2.txt
02. sending incremental file list
03. ./
04. 2.txt //传输文档列表
- 05.
06. sent 100 bytes received 34 bytes 268.00 bytes/sec
07. total size is 0 speedup is 0.00

确认目录1和目录2的内容已经一致：

```
01. [root@svr7 ~]# ls /fdir/ /tdir/
02. /fdir/:
03. 1.txt 2.txt
04.
05. /tdir/:
06. 1.txt 2.txt
```

再次跟踪同步信息，已经无需传输文件：

```
01. [root@svr7 ~]# rsync -av /fdir/ /tdir
02. sending incremental file list
03.
04. sent 58 bytes received 12 bytes 140.00 bytes/sec
05. total size is 0 speedup is 0.00
```

### 3) 验证-n选项

将-n、-v选项合用，可以模拟同步过程，显示需要做哪些操作（但并不真的同步）。

在目录/fdir下新建文件3.txt，测试同步操作：

```
01. [root@svr7 ~]# touch /fdir/3.txt
02. [root@svr7 ~]# rsync -avn /fdir/ /tdir/
03. sending incremental file list
04. ./
```

[Top](#)

- 05. 3.txt //提示同步时会传输哪些文件
- 06.
- 07. sent 78 bytes received 18 bytes 192.00 bytes/sec
- 08. total size is 0 speedup is 0.00 ( DRY RUN)
- 09. [ root@svr7 ~] # ls -l /tdir/3.txt //但实际并未真的同步
- 10. ls: 无法访问/tdir/3.txt: 没有那个文件或目录

去掉-n选项才会真正同步：

- 01. [ root@svr7 ~] # rsync -av /fdir/ /tdir/
- 02. sending incremental file list
- 03. ./
- 04. 3.txt
- 05.
- 06. sent 114 bytes received 34 bytes 296.00 bytes/sec
- 07. total size is 0 speedup is 0.00
- 08. [ root@svr7 ~] # ls -l /tdir/3.txt
- 09. -rw-r--r--. 1 root root 0 1月 11 21: 46 /tdir/3.txt

#### 4 ) 验证--delete选项

rsync同步操作默认只是将目录1的数据同步到目录2，但如果目录2存在多余的文件却并不会去除，除非添加—delete选项。

在目录/fdir、/tdir已经完成同步后，删除/tdir/2.txt文件，再次同步：

- 01. [ root@svr7 ~] # rm -rf /fdir/2.txt
- 02. [ root@svr7 ~] # rsync -a /fdir/ /tdir/

[Top](#)

检查发现目标文件夹/tdir下的2.txt文件还在：

```
01. [root@svr7 ~]# ls /fdir/ /tdir/
02. /fdir/:
03. 1.txt 3.txt
04.
05. /tdir/:
06. 1.txt 2.txt 3.txt
```

这种情况下添加--delete选项再次执行同步，两个目录的内容就一致了：

```
01. [root@svr7 ~]# rsync -a --delete /fdir/ /tdir/
02. [root@svr7 ~]# ls /fdir/ /tdir/
03. /fdir/:
04. 1.txt 3.txt
05.
06. /tdir/:
07. 1.txt 3.txt
```

## 2 案例2：rsync+SSH同步

### 2.1 问题

本例要求掌握rsync与远程SSH资源的同步操作，使用rsync命令访问远程主机svr7，完成下列任务：

[Top](#)

1. 查看远程主机的 / 目录下有哪些子目录
2. 从远程主机下载 /etc/passwd 文件到当前目录



3. 将远程主机的 /boot/ 目录同步为本地的 /fromssh
4. 将本机的 /etc 目录同步到远程主机的 /opt/下

## 2.2 方案

列出 SSH 服务端资源

- rsync user@host:远程目录/

rsync+SSH远程同步操作：

- rsync [...] user@host:远程目录 本地目录
- rsync [...] 本地目录 user@host:远程目录

## 2.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：列出远程主机的SSH资源

查看远程主机svr7的/目录下有哪些子目录：

```
01. [root@pc207 ~]# rsync root@192.168.4.7: /
02. root@192.168.4.7's password:           //验证对方的密码
03. dr- xr- xr- x      4096 2016/12/15 10: 39: 34 .
04. lrwxrwxrwx        7 2016/12/07 09: 21: 50 bin
05. lrwxrwxrwx        7 2016/12/07 09: 21: 50 lib
06. lrwxrwxrwx        9 2016/12/07 09: 21: 50 lib64
07. lrwxrwxrwx        8 2016/12/07 09: 21: 50/sbin
08. dr- xr- xr- x      4096 2016/12/07 11: 25: 29 boot
09. drwxr- xr- x       6 2016/12/07 09: 21: 14 data
10. drwxr- xr- x      3200 2016/12/15 10: 46: 15 dev
11. drwxr- xr- x      8192 2016/12/20 17: 01: 02 etc
```

[Top](#)

## 步骤二：rsync+SSH同步操作

1 ) 从远程主机svr7下载/etc/passwd文件到当前目录

```
01. [root@pc207 ~]# rsync root@192.168.4.7:/etc/passwd ./
02. root@192.168.4.7's password:           //验证对方的密码
03. [root@pc207 ~]# cat passwd             //检查同步结果
04. root:x:0:0:root:/root:/bin/bash
05. bin:x:1:1:bin:/bin:/sbin/nologin
06. daemon:x:2:2:daemon:/sbin:/sbin/nologin
07. adm:x:3:4:adm:/var/adm:/sbin/nologin
08. lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
09. ...
```

2 ) 将远程主机svr7的/boot/目录同步为本地的/fromssh

```
01. [root@pc207 ~]# rsync -a root@192.168.4.7:/boot/ /fromssh
02. root@192.168.4.7's password:           //验证对方的密码
03. [root@pc207 ~]# ls /fromssh/           //检查同步结果
04. config-3.10.0-327.el7.x86_64
05. extlinux
06. grub2
07. initramfs-0-rescue-a19921505cc7e19d20dfcd5cea7d8aa2.img
08. initramfs-3.10.0-327.el7.x86_64.img
09. initramfs-3.10.0-327.el7.x86_64kdump.img
10. ...
```

[Top](#)

3) 将本机的/etc目录同步到远程主机svr7的/opt/下

确认目录大小：

```
01. [root@pc207 ~]# du -sh /etc
02. 35M /etc
```

上行同步到远程主机svr7上：

```
01. [root@pc207 ~]# rsync -a /etc root@192.168.4.7:/opt/
02. root@192.168.4.7's password:
```

在远程主机上检查同步结果：

```
01. [root@svr7 ~]# du -sh /opt/etc
02. 35M /opt/etc
```

## 3 案例3：配置rsync服务端

### 3.1 问题

本例要求在主机svr7上配置rsync同步资源，完成下列任务：

1. 将 /usr/src 目录配置为 rsync 共享，其共享名为 tools，仅允许用户 ruser 以密码 pwd123 访问
2. 启动 rsync 服务，并将其设为开机自启

[Top](#)

## 3.2 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：建立rsync账号文件

#### 1) 建立账号记录

```
01. [root@svr7 ~] # vim /etc/rsyncd_users.db
02. ruser:pwd123           //每行一个用户记录
03. othername:123456
```

#### 2) 控制账号文件权限（服务端严格模式有要求）

```
01. [root@svr7 ~] # chmod 600 /etc/rsyncd_users.db
02. [root@svr7 ~] # ls -l /etc/rsyncd_users.db //确认权限调整结果
03. -rw----- . 1 root root 27 12月 20 16:25 /etc/rsyncd_users.db
```

### 步骤二：配置rsync共享资源

#### 1) 确保被共享的目录可用

```
01. [root@svr7 ~] # ls /usr/src/ //目录及文档资源存在
02. copyright debug install.sh license_ch.txt src
03. data inotify-tools 3.13 kernels README.md
04.
05. [root@svr7 ~] # ls -ld /usr/src/ //权限允许访问
06. drwxr-xr-x. 7 root root 4096 1月 7 13:57 /usr/src/
```

[Top](#)

## 2) 建立rsyncd.conf配置文件

```
01. [root@svr7 ~]# vim /etc/rsyncd.conf
02. [tools]                                     //共享名
03. path = /usr/src                           //目录位置
04. comment = Rsync Test Directory
05. read only = yes                           //默认只读
06. dont compress = *.gz *.bz2 *.tgz *.zip    //不需要压缩的文件
07. auth users = ruser                        //允许谁访问
08. secrets file = /etc/rsyncd_users.db       //账号文件
```

## 步骤三：启用rsync服务端

### 1) 启动系统服务rsyncd，并设置开机自启

```
01. [root@svr7 ~]# systemctl restart rsyncd
02. [root@svr7 ~]# systemctl enable rsyncd
03. Created symlink from /etc/systemd/system/multi-user.target.wants/rsync.service to /usr/lib/systemd/system/rsync.service.
```

### 3) 检查服务监听状态

```
01. [root@svr7 ~]# netstat -anptu | grep rsync
02. tcp      0      0 0.0.0.0:873      0.0.0.0:*        LISTEN      6339/rsync
03. tcp6     0      0 :::873          :::*             LISTEN      6339/rsync
```

[Top](#)

## 4 案例4：访问rsync共享资源

### 4.1 问题

本例要求掌握rsync与远程rsync资源的同步操作，使用rsync命令访问远程主机svr7，完成下列任务：

1. 列出远程主机提供的 rsync 资源，并查看内容
2. 将远程主机的 rsync 共享目录同步到本机

### 4.2 方案

列出 rsync 共享资源：

- rsync host::
- rsync rsync://host

rsync+rsync远程同步操作：

- rsync [...] user@host::共享名 本地目录
- rsync [...] 本地目录 user@host::共享名

访问需要验证的rsync资源时，可以通过 --password-file= 选项来加载提前准备的密码文件（权限600，不要包括除密码外的其他字符），便于计划任务或脚本使用。

### 4.3 步骤

实现此案例需要按照如下步骤进行。

**步骤一：列出远程主机提供的 rsync 资源，并查看内容**

1 ) 列出有哪些rsync共享（无需密码）

使用方式1：

```
01. [root@pc207 ~]# rsync 192.168.4.7::
02. tools          Rsync Test Directory
```

[Top](#)

或者，使用方式2：

01. [ root@pc207 ~] # rsync rsync://192.168.4.7
02. tools Rsync Test Directory

2) 列出指定的rsync共享目录下有哪些内容

访问目录内容时要求用户验证，否则会失败：

01. [ root@pc207 ~] # rsync 192.168.4.7 : : tools/
02. Password:
03. @ERROR: auth failed on module tools
04. rsync error: error starting client-server protocol (code 5) at main.c(1516) [Receiver=3.0.9]

指定正确的用户名、密码才能够访问：

01. [ root@pc207 ~] # rsync ruser@192.168.4.7 : : tools/
02. Password: //输入用户ruser的密码
03. drwxr-xr-x 4096 2017/01/07 13:57:06 .
04. -rw-r--r-- 303 2015/12/23 19:36:55 README.md
05. -rw-r--r-- 2189 2015/11/18 20:47:12 copy right
06. -rwxr-xr-x 773 2015/11/18 20:47:12 install.sh
07. -rw-r--r-- 7828 2015/11/18 20:47:12 license\_ch.txt
08. drwxr-xr-x 4096 2017/01/07 13:51:21 data

[Top](#)

## 步骤二：将远程主机的 rsync 共享目录同步到本机

### 1) 手动交互方式

将远程tools共享的资源同步到本机的/mysrc目录：

```
01. [root@pc207 ~]# rsync -az ruser@192.168.4.7::tools/ /my src
02. Password: //输入用户ruser的密码
03. [root@pc207 ~]# du -sh /my src //检查同步结果
04. 17M /my src
```

### 2) 自动交互方式

通过--password-file选项提供密码文件：

```
01. [root@pc207 ~]# echo pwd123 > /root/pass.txt //建立密码文件
02. [root@pc207 ~]# chmod 600 /root/pass.txt //控制密码文件的权限
03. [root@pc207 ~]# rsync -az --password-file=/root/pass.txt ruser@192.168.4.7::tools/ /my src
04. //免除手动密码交互
```

## 5 案例5：使用inotifywait工具

### 5.1 问题

本例要求安装inotify-tools工具，并针对文件夹 /opt 启用 inotifywait 监控，完成下列任务：

[Top](#)

1. 当此目录下出现新建、修改、更改权限、删除文件等事件时能给出提示



## 2. 验证上述监控事件的效果

## 5.2 方案

inotifywait监控操作：

- inotifywait [选项] 目标文件夹

inotifywait常用命令选项：

- -m，持续监控（捕获一个事件后不退出）
- -r，递归监控、包括子目录及文件
- -q，减少屏幕输出信息
- -e，指定监视的 modify、move、create、delete、attrib 等事件类别

## 5.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：安装inotify-tools软件包

#### 1) 解包

```
01 [root@svr7 ~]# tar xf inotify-tools-3.13.tar.gz -C /usr/src/
```

#### 2) 配置

```
01 [root@svr7 ~]# cd /usr/src/inotify-tools-3.13/
```

```
02 [root@svr7 inotify-tools-3.13]# ./configure
```

[Top](#)

#### 3) 编译

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

#### 4 ) 安装

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

#### 5 ) 检查安装结果 ( inotifywait程序可用 )

```
01. [ root@svr7 ~] # inotifywait -- help
02. inotifywait 3.13
03. Wait for a particular event on a file or set of files.
04. Usage: inotifywait [ options ] file1 [ file2 ] [ file3 ] [ ... ]
05. Options:
06.     -h|--help      Show this help text.
07.     ... ..
```

### 步骤二：测试inotifywait监控

#### 1 ) 开启监控任务，置入后台

```
01. [ root@svr7 ~] # inotifywait -mrq -e create,modify,move,attrib,delete /opt &
02. [ 1] 55564
```

[Top](#)

#### 2 ) 测试/opt/目录下的新建、修改、改名、更改权限、删除文件等事件的响应消息

观察新建文件时的监控信息：

- 01. [ root@svr7 ~] # touch /opt/a.txt
- 02. /opt/ CREATE a.txt
- 03. /opt/ ATTRIB a.txt

观察修改文件内容时的监控信息：

- 01. [ root@svr7 ~] # echo Hello > /opt/a.txt
- 02. [ root@svr7 ~] # /opt/ MODIFY a.txt
- 03. /opt/ MODIFY a.txt

观察将文件改名时的监控信息：

- 01. [ root@svr7 ~] # mv /opt/a.txt /opt/b.txt
- 02. /opt/ MOVED\_FROM a.txt
- 03. /opt/ MOVED\_TO b.txt

观察修改文件权限时的监控信息：

- 01. [ root@svr7 ~] # chmod 600 /opt/b.txt
- 02. /opt/ ATTRIB b.txt

[Top](#)

观察删除文件时的监控信息：

- 01. [ root@svr7 ~] # rm -rf /opt/b.txt
- 02. /opt/ DELETE b.txt

3 ) 停止监控任务

- 01. [ root@svr7 ~] # kill -9 %1
- 02. [ 1] + 已杀死 inotifywait -mr -e create,modify,move,attrib,delete /opt

## 6 案例6：配置Web镜像同步

### 6.1 问题

本例要求为两台Web服务器svr7、pc207的网页文档目录配置镜像同步，主要基于inotifywait监控技术实现实时触发操作，需要完成下列任务：

- 1. 以 svr7 为发起方，原始目录为 /var/www/html/
- 2. 以 pc207 为同步目标，基于SSH免密验证
- 3. 编写 inotify+rsync 同步脚本，验证实时同步效果

### 6.2 方案

inotifywait与rsync的结合，主要思路：

- 01. while inotifywait监控操作
- 02. do
- 03. 需要执行的rsync同步操作
- 04. done

[Top](#)

## 6.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：为主机svr7、pc207部署同步目录

双方的目录均为/var/www/html/，如果安装了httpd，此目录会自动出现。

#### 1) 确认svr7的目录内容

```
01. [root@svr7 ~]# yum -y install httpd
02. ...
03. [root@svr7 ~]# ls /var/www/html/           //向目录下提供一些测试文件
04.      libreoffice
```

#### 2) 确认pc207的目录内容

```
01. [root@pc207 ~]# yum -y install httpd
02. ...
03. [root@pc207 ~]# ls /var/www/html           //初始目录无数据
04. [root@pc207 ~]#
```

### 步骤二：为svr7配置到pc207的SSH密钥对验证，实现免密码交互

#### 1) 检查当前用户是否已经有可用的SSH密钥对文件

```
01. [root@svr7 ~]# ls ~/.ssh/id_*
```

[Top](#)

02.     /root/.ssh/id\_rsa /root/.ssh/id\_rsa.pub

如果找不到id\_rsa、id\_rsa.pub密钥对文件，则需要执行下列操作创建：

```
01. [ root@svr7 ~] # ssh-keygen
02. Generating public/private rsa key pair.
03. Enter file in which to save the key (/root/.ssh/id_rsa):    //按回车，确认存放位置
04. Enter passphrase (empty for no passphrase):    //按回车，确认不要密码
05. Enter same passphrase again:    //再次按回车，确认
06. Your identification has been saved in /root/.ssh/id_rsa.
07. Your public key has been saved in /root/.ssh/id_rsa.pub.
08. The key fingerprint is:
09. 00:a7:cb:2d:9d:b8:8a:df:f5:ff:5b:ed:bd:04:10:fe root@svr7
10. The key's randomart image is:
11. +- [ RSA 2048]-----+
12. |  ..  .  |
13. |  +  ..  |
14. |  ..  o  |
15. |  . = o  o  |
16. |  = + S  E  |
17. |  o      .. |
18. |  ..     ...|
19. |  . o . .  ....|
20. | .. o .  ....o. .+|
21. +-----+
```

[Top](#)

## 2 ) 将当前用户的SSH公钥部署到远程主机

```
01. [ root@svr7 ~] # ssh-copy-id root@192.168.4.207
02. The authenticity of host '192.168.4.207 (192.168.4.207)' can't be established.
03. ECDSA key fingerprint is d3:16:2c:9a:9d:91:28:c8:74:9c:af:2d:04:82:c9:66.
04. Are you sure you want to continue connecting (yes/no)? yes //首次连yes确认
05. root@192.168.4.207's password: //验证对方的密码
06.
07. Number of key(s) added: 1
08.
09. Now try logging into the machine, with: "ssh 'root@192.168.4.207'"
10. and check to make sure that only the key(s) you wanted were added.
```

## 3 ) 验证免密码登录效果

```
01. [ root@svr7 ~] # ssh root@192.168.4.207
02. Last login: Fri Jan 13 09:52:08 2017 from 192.168.4.110
03. [ root@pc207 ~] # //确认已免密码连入远程主机
04. [ root@pc207 ~] # exit //退出SSH登录环境
05. 登出
06. Connection to 192.168.4.207 closed.
07. [ root@svr7 ~] # //已反对原客户机
```

## 步骤三：编写镜像同步脚本并测试效果

[Top](#)

### 1 ) 编写脚本文件/root/isync.sh

```
01. [root@svr7 ~] # vim /root/isync.sh
02. #!/bin/bash
03. FROM_DIR="/var/www/html/"
04. RSYNC_CMD="rsync -az --delete $FROM_DIR root@192.168.4.207: /var/www/html"
05. while inotifywait -rqq -e modify,move,create,delete,attrib $FROM_DIR
06. do
07.     $RSYNC_CMD
08. done &
09. [root@svr7 ~] # chmod +x /root/isync.sh
```

## 2 ) 运行脚本

```
01. [root@svr7 ~] # /root/isync.sh
02. [root@svr7 ~] # pgrep -l inotify           //确认任务在运行
03. 56494 inotifywait
```

## 3 ) 测试同步效果

在svr7上向/var/www/html/目录下添加一个测试网页（触发同步）：

```
01. [root@svr7 ~] # touch /var/www/html/a.txt
02. [root@svr7 ~] # ls /var/www/html/
03. a.txt libreoffice
```

[Top](#)

在pc207上检查/var/www/html/目录，内容应该已经与svr7上的同名目录一致：



```
01. [root@pc207 ~]# ls /var/www/html
02. a.txt libreoffice
```

4) 结束测试后，在svr7上停止监控任务

```
01. [root@svr7 ~]# pkill -9 inotify
02. [root@svr7 ~]# pgrep -l inotify           //确认已没有监控任务
03. [root@svr7 ~]#
```

## 7 案例7：配置并验证Split分离解析

### 7.1 问题

本例要求配置一台智能DNS服务器，针对同一个FQDN，当不同的客户机来查询时能够给出不同的答案。需要完成下列任务：

1. 从主机192.168.4.207查询时，结果为：www.tedu.cn ---> 192.168.4.100
2. 从其他客户端查询时，www.tedu.cn ---> 1.2.3.4

### 7.2 方案

在配置DNS服务器时，通过view视图设置来区分不同客户机、不同地址库：

```
01. view "视图1" {
02.     match-clients { 客户机地址1; ...; };           //匹配第1类客户机地址
03.     zone "目标域名" IN {                             //同一个DNS区域
04.         type master;
05.         file "地址库1";                               //第1份地址库
```

[Top](#)

```

06.     };
07. };
08. view "视图2" {
09.     match-clients { 客户机地址2; ... }; //匹配第2类客户机地址
10.     match-clients { any; }; //匹配任意地址
11.     zone "目标域名" IN { //同一个DNS区域
12.         type master;
13.         file "地址库2"; //第2份地址库
14.     };
15. };
16. ...
17. view "视图n" {
18.     match-clients { any; }; //匹配任意地址
19.     zone "目标域名" IN { //同一个DNS区域
20.         type master;
21.         file "地址库n"; //第n份地址库
22.     };
23. };

```

## 7.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：配置Split分离解析

1) 为tedu.cn区域建立两份解析记录文件

第一份解析记录文件提供给客户机192.168.4.207、网段192.168.7.0/24，对应目标域名www.tedu.cn的A记录地址为192.168.4.100。相关操作及配置如下：

[Top](#)

```
01. [root@svr7 ~]# cd /var/named/
02. [root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.lan
03. [root@svr7 named]# vim tedu.cn.zone.lan
04. $TTL 1D
05. @ IN SOA @ rname.invalid. (
06.         0 ; serial
07.         1D ; refresh
08.         1H ; retry
09.         1W ; expire
10.        3H ) ; minimum
11. @ NS svr7.tedu.cn.
12. svr7 A 192.168.4.7
13. pc207 A 192.168.4.207
14. www A 192.168.4.100
```

第二份解析记录文件提供给其他客户机，对应目标域名www.tedu.cn的A记录地址为1.2.3.4。相关操作及配置如下：

```
01. [root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.other
02. [root@svr7 named]# vim tedu.cn.zone.other
03. $TTL 1D
04. @ IN SOA @ rname.invalid. (
05.         0 ; serial
06.         1D ; refresh
07.         1H ; retry
08.         1W ; expire
09.        3H ) ; minimum
```

[Top](#)

```
10. @ NS svr7.tedu.cn.
11. svr7 A 192.168.4.7
12. pc207 A 192.168.4.207
13. www A 1.2.3.4
```

2 ) 修改named.conf配置文件，定义两个view，分别调用不同解析记录文件

```
01. [ root@svr7 ~]# vim /etc/named.conf
02. options {
03.     directory "/var/named";
04. };
05. acl "my lan" {                                //名为my lan的列表
06.     192.168.4.207; 192.168.7.0/24;
07. };
08. ...
09. view "my lan" {
10.     match-clients { my lan; };                //检查客户机地址是否匹配此列表
11.     zone "tedu.cn" IN {
12.         type master;
13.         file "tedu.cn.zone.lan";
14.     };
15. };
16. view "other" {
17.     match-clients { any; };                    //匹配任意客户机地址
18.     zone "tedu.cn" IN {
19.         type master;
20.         file "tedu.cn.zone.other";
```

[Top](#)

21.     };
22.     };

### 3 ) 重启named服务

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

### 步骤二：测试分离解析效果

#### 1 ) 从mylan地址列表中的客户机查询

在客户机192.168.4.207 ( 或网段192.168.7.0/24内的任意客户机 ) 上查询www.tedu.cn , 结果是 192.168.4.100 :

```
01.     [ root@pc207 ~] # host www.tedu.cn 192.168.4.7
02.     Using domain server:
03.     Name: 192.168.4.7
04.     Address: 192.168.4.7#53
05.     Aliases:
06.
07.     www.tedu.cn has address 192.168.4.100
```

#### 2 ) 从其他客户机查询

在DNS服务器本机或CentOS真机上查询www.tedu.cn时 , 结果为 1.2.3.4 :

[Top](#)

```
01.     [ root@svr7 ~] # host www.tedu.cn 192.168.4.7
02.     Using domain server:
```

- 03. Name: 192.168.4.7
- 04. Address: 192.168.4.7#53
- 05. Aliases:
- 06.
- 07. www.tedu.cn has address 1.2.3.4