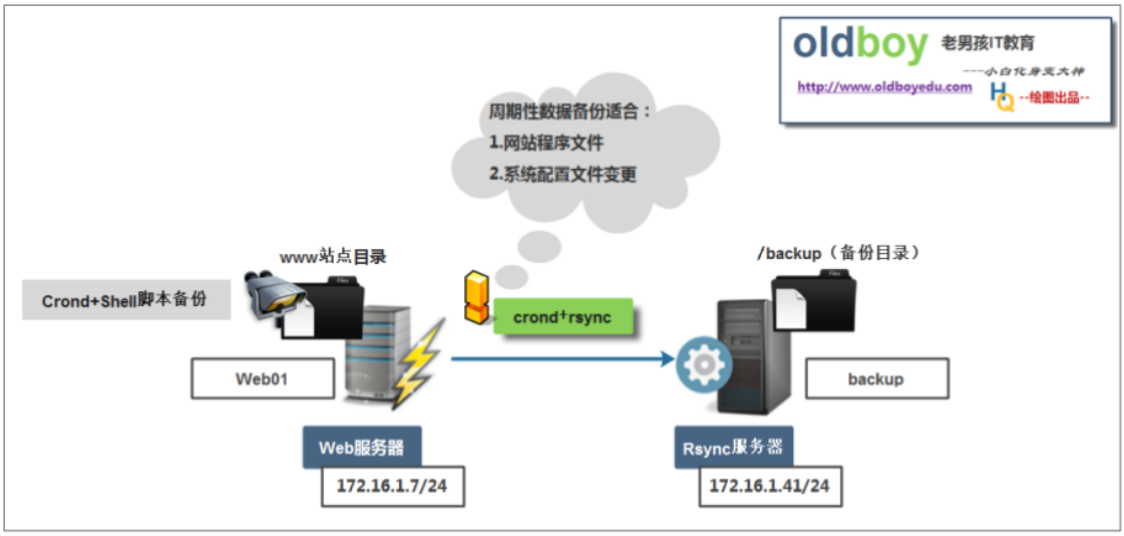
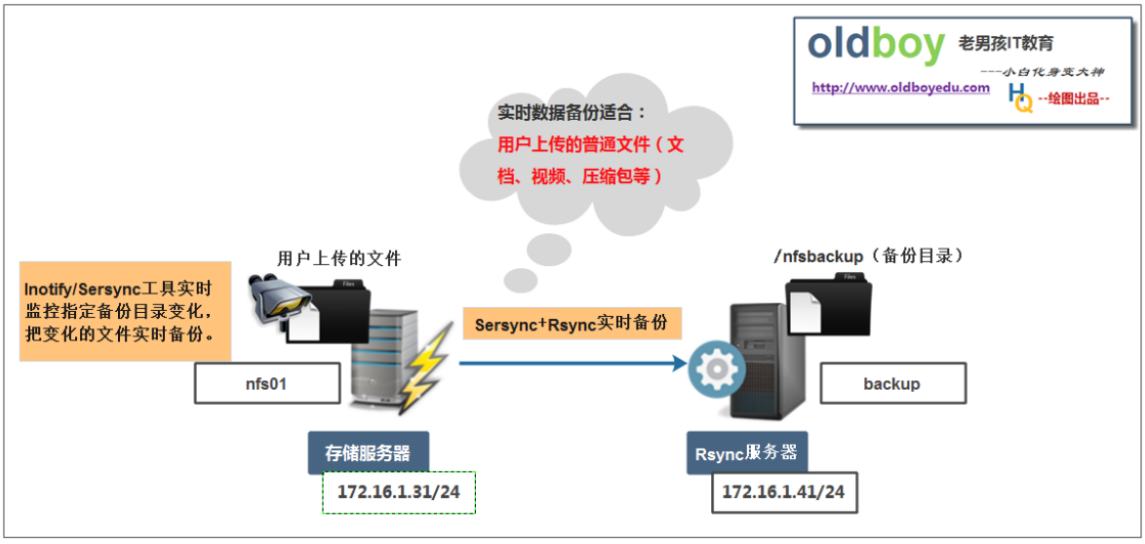
Day43-课堂笔记-老男孩Linux77期-集群架构-inotify-sersync-实时复制

1）企业级备份方案介绍

1.利用定时方式，实现周期备份重要数据信息。



2.实时数据备份方案



3.实时复制环境准备

主机名称	角色	IP(eth1)	目录信息
nfs01	NFS 服务器端	172.16.1.31	/data
backup	Rsync备份服务器	172.16.1.41	/backup

4.实时复制软件介绍

随着互联网技术的不断发展，数据复制软件服务也层出不穷，目前企业中较为常用的实时复制软件有如下两种列表显示：

表4-2 实时复制软件对比

软件服务	依赖程序	部署难点	说明
inotify-tools	rsync守护进程服务	复制脚本编写	监控目录数据变化，实时数据复制
sersync*	rsync守护进程服务	*配置*文件编写	监控目录数据变化，实时数据复制

inotify-tools #原始的监控软件。sersync #封装了intotify，增加了一些功能。

5.实时复制inotify机制介绍

6.项目部署实施

1）部署环境准备

确保rsync服务端配置完成

确保远程数据**传输服务部署完成**

```
[root@nfs01 data]# rsync -avz  /data
rsync_backup@172.16.1.41::backup --password-
file=/etc/rsync.passwordsending incremental file list
data/
data/oldboy-2.ev4
data/潘晓婷.jpg
data/花瓶美女1.jfif
data/花瓶美女2.jfif
data/老男孩Linux77期视频/
data/老男孩Linux77期视频/02-老男孩77期-什么是集群？.ev4

sent 51,802,259 bytes  received 135 bytes  20,720,957.60
bytes/sec
total size is 52,144,218  speedup is 1.01
```

```
[root@nfs01 data]#
[root@nfs01 data]# rsync -avz  /data
rsync_backup@172.16.1.41::backup --password-
file=/etc/rsync.passwordsending incremental file list

sent 310 bytes  received 26 bytes  672.00 bytes/sec
total size is 52,144,218  speedup is 155,191.12
```

2)检查Linux系统支持inotify实时监控

```
[root@nfs01 data]# uname -r
3.10.0-1160.31.1.el7.x86_64
[root@nfs01 data]# ls -l /proc/sys/fs/inotify/
总用量 0
-rw-r--r-- 1 root root 0 7月   2 17:07 max_queued_events
-rw-r--r-- 1 root root 0 7月   2 17:07 max_user_instances
-rw-r--r-- 1 root root 0 7月   2 17:07 max_user_watches
[root@nfs01 data]# cat
/proc/sys/fs/inotify/max_queued_events
16384
[root@nfs01 data]# cat
/proc/sys/fs/inotify/max_user_instances
128
[root@nfs01 data]# cat
/proc/sys/fs/inotify/max_user_watches
8192
```

文件名称	作用说明
max_user_watches:	设置inotify命令可以监视的文件数量（单进程）。
max_user_instances	设置每个用户可以运行的inotify命令的进程数。
max_queued_events	设置inotify实例事件（event）队列可容纳的事件数量。

3)inotify-tools工具安装过程

安装inotify-tools工具比较简单，执行的命令过程如下：

```
[root@nfs01 ~]# yum install epel-release -y          #<==需先安装epel，默认官方没有。
[root@nfs01 ~]# yum install inotify-tools -y         #<==下载inotify软件工具。
[root@nfs01 ~]# rpm -ql inotify-tools|head -2
/usr/bin/inotifywait     #<==inotifywait软件命令（重要）。
/usr/bin/inotifywatch    #<==inotifywatch软件命令（重要）。
```

□inotifywait：在被监控的目录等待特定文件系统事件（open、close、delete等）发生，执行后处于阻塞状态，适合在Shell脚本中使用，此命令是实现监控的重点。

□inotifywatch：收集被监控的文件系统使用的统计数据，指文件系统事件发生的次数统计。

4)**inotifywait命令参数

命令参数	参数说明
-m|--monitor	始终保持事件监听状态(重要参数)
-d|--daemon	类似于-m参数，只是将命令运行在后台记录触发的事件信息在指定文件中，利用--outfile参数定义程序日志使用--syslog参数
-r	递归监控目录数据信息变化(重要参数)
-o|--outfile	打印事件到文件中，相当于标准正确输出
-s|--syslog	发送错误到syslog相当于标准错误输出。
-q|--quiet	输出信息少（只打印事件信息）
--excludei	排除文件或目录时，不区分大小写
--timefmt	指定时间输出的格式
--format	打印使用指定的输出类似格式字符串；即实际监控输出的内容
-e	指定监听指定的事件，如果省略，表示所有事件都进行监听。**(重要*****参数)**

5)监控事件说明

常用监控文件事件功能参数

事件名称	事件说明
access	文件或目录内容被读取
modify	文件或目录内容被写入
attrib	文件或目录属性改变
close_write	文件或目录关闭，在写入模式打开之后关闭的。(重要参数)
close_nowrite	文件或目录关闭，在只读模式打开之后关闭的
close	文件或目录关闭，不管读或是写模式
open	文件或目录被打开（重要）
moved_to	文件或目录被移动到监控的目录中
moved_from	文件或目录从监控的目录中被移动
move	文件或目录不管移动到或是移动出监控目录都触发事件
create	文件或目录创建在监控的目录中**(重要参数)**
delete	文件或目录被删除在监控的目录中**(重要参数)**
delete_self	文件或目录被删除
unmount	文件系统包含的文件或目录不能卸载

6）测试事件

**

```
q** **测试create****事件**
**#** **在NFS服务器****上开启inotify相应监控功能**
[root@nfs01 ~]# inotifywait -mrq --timefmt '%d/%m/%y
%H:%M' --format '%T %w%f' -e create /data #<==监控/data目录
创建事件显示信息。
14/04/19 19:26 /data/a.txt #<==目录随着下面的操作，文件实时出
现。
14/04/19 19:26 /data/b.txt
```

```
**#** **再****开启一个****NFS服务器****连接窗口进行测试验证相应****事件**
[root@nfs01 ~]# touch /data/a.txt
[root@nfs01 ~]# touch /data/b.txt
```

说明：只监控create创建事件，其它事件并没有进行监控，其它事件产生，监控服务没有信息输出

□测试delete事件

```
# 在NFS服务器上开启inotify相应监控功能。
[root@nfs01 data]# inotifywait -mrq -e delete /data
/data/ DELETE 1.txt
/data/ DELETE 2.txt
/data/ DELETE 3.txt

[root@nfs01 data]# inotifywait -mrq -e delete,close_write /data
/data/ CLOSE_WRITE,CLOSE oldboy.txt
/data/ CLOSE_WRITE,CLOSE oldboy.txt
/data/ DELETE oldboy.txt
```

7)inotify-tools软件部署项目实战

 # 在NFS服务器上开启inotify相应监控功能。**

```
[root@nfs01 ~]# inotifywait -mrq  -e close_write,delete /data  #<==监控/data目录增删改事件。

# 编写实时监控和复制脚本
```

本例涉及while循环脚本语法示例：

```
[root@nfs01 ~]# seq 3 >test.txt #<==生成3个数字，每行一个到test.txt文件里。
[root@nfs01 ~]# cat test.txt    #<==查看结果。
1
2
3
[root@nfs01 ~]# cat w.sh  #<==开发脚本。
cat ./test.txt|\   #<==按行读取test.txt文件，将每一行赋值给下文的$line变量。
while read line    #<==循环读取line变量的内容。
do
   echo $line      #<==打印line变量的内容。
done               #<==读取完test.txt内容后才会终止循环。
[root@nfs01 ~]# sh -x w.sh    #<==查看执行原理过程。
```

```
+ read line
+ cat ./test.txt    #<==按行读取test.txt文件，将第一行赋值给
line。
+ echo 1            #<==输出1。
1
+ read line         #<==按行读取test.txt文件，将第二行赋值给
line。
+ echo 2            #<==输出2。
2
+ read line         #<==按行读取test.txt文件，将第三行赋值给
line。
+ echo 3            #<==输出3。
3
+ read line
# 有关Shell知识，读者可以学习《跟老男孩学习Linux运维:Shell编程实战
一书》
```

8)生产脚本（nfs01）

```
#!/bin/sh
cmd="/usr/bin/inotifywait"
$cmd -mrq  -e close_write,delete /data|\
while read line
do
  cd /data&&\
  rsync -az --delete ./ rsync_backup@172.16.1.41::backup -
-password-file=/etc/rsync.password
done
#sh -x 脚本名  #<==调试脚本
```

nfs01上执行脚本

```
/bin/sh /server/scripts/monitoy1.sh &
touch /data/{1..10}.txt
```

backup服务器上检查

```
ls /backup
```

nfs01上执行：

测试没有问题后，在nfs01上让脚本在后台运行，并放入/etc/rc.local里。

```
[root@nfs01 scripts]# /bin/sh /server/scripts/monitor.sh
&>/dev/null & #<==&表示让脚本在后台运行。
[root@nfs01 scripts]# tail -2 /etc/rc.local
#inotify sync data by oldboy at 201808
/bin/sh /server/scripts/monitor.sh &>/dev/null &
                #<==放到开机自启动文件里。
```

说明：让脚本在后台运行

9)inotify-tools软件优化企业案例

利用inotify对应的proc目录中的三个文件，可以适当对inotify软件进行优化，企业实战中调整为如下配置：

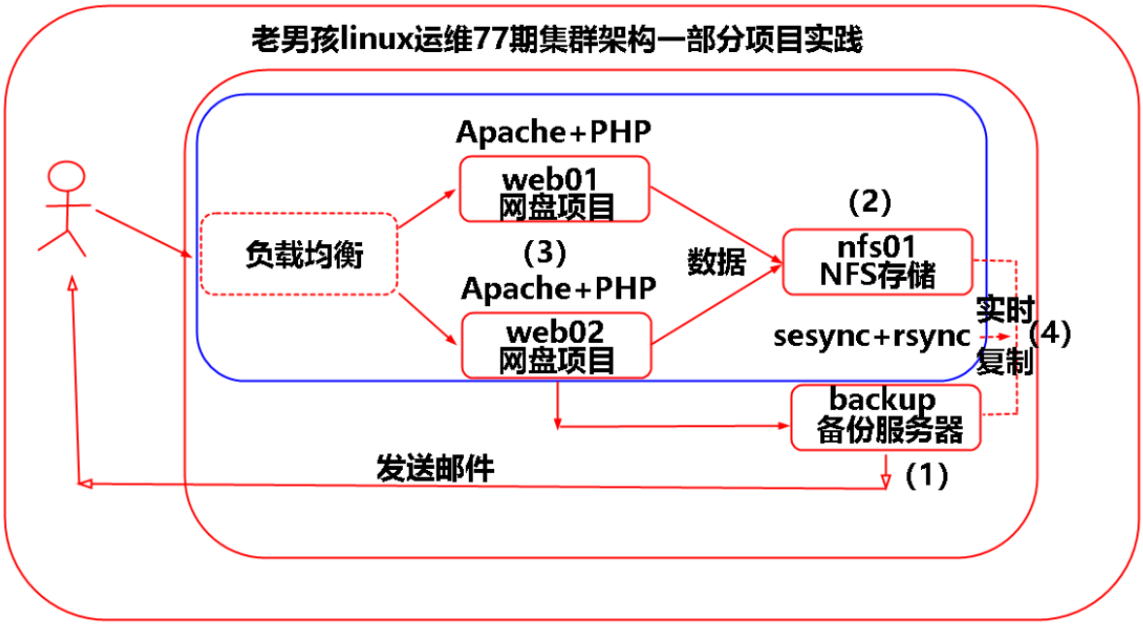
```
[root@nfs01 scripts]#  echo "50000000"
>/proc/sys/fs/inotify/max_user_watches    #<==单进程可以监控
的文件数量。
[root@nfs01 scripts]#  echo "50000000"
>/proc/sys/fs/inotify/max_queued_events  #<==队列容纳事件的
数量。
# 说明：以上配置重启可能会失效，需要放置在rc.local文件中。
# 设置开机自动进行实时监控优化。

[root@nfs01 data]# tail -5 /etc/rc.local
#inotiy-tools 命令
/bin/sh /server/scripts/monitoy1.sh &
######inotify 优化
echo "50000000" >/proc/sys/fs/inotify/max_user_watches
echo "50000000" >/proc/sys/fs/inotify/max_queued_events
```

课后作业：

1）sersync实时复制工具

2）下图项目搞定。


老男孩linux运维77期集群架构一部分项目实践
Apache+PHP
web01
网盘项目
(2)
负载均衡
(3)
数据
nfs01
NFS存储
Apache+PHP
web02
网盘项目
sesync+rsync
实时
复制
(4)
backup
备份服务器
发送邮件
(1)