



原创

# rsync+ Notify配置解析及步骤详解



上古杰作

[关注](#)

2018-08-13 17:29:43 1263人阅读 0人评论

## rsync步骤配置

### Rsync介绍

#### 什么是rsync

rsync是一款开源，快速，多功能的可实现增量的本地或远程的数据镜像同步备份的优秀工具。适用于多个平台。从软件名称可以看出是远程同步的意思（remote sync）。可使本地主机不同分区或目录之间及本地和远程两台主机之间的数据快速同步镜像，远程备份等功能。

在同步备份时，默认情况下，rsync通过其独特的“quick check”算法，仅同步大小或者最后修改时间发生变化的文件或目录（也可根据权限，属主等变化同步，需要制定参数）。甚至是只同步一个文件里变化的内容部分，所以可以实现快速的同步数据的功能。

提示：传统的cp，scp工具拷贝每次均为完整拷贝，而rsync除了完整拷贝，还具备增量拷贝的功能，因此从此性能及效率上更胜一筹。

#### Rsync的特性如下：

- 1) 支持拷贝特殊文件如链接，设备等
- 2) 可以有排除指定文件或目录同步的功能，相当于打包命令tar
- 3) 可以保持原来文件或目录的权限，时间，软硬链接等所有属性均不改变。
- 4) 可实现增量同步，即只同步发生变化的数据，因此数据传输效率更高
- 5) 可以使用rcp，rsh，ssh等方式来配合传输文件，也可以通过直接的socket链接
- 6) 支持匿名的或认证的进程模式传输，方便进行数据备份及镜像。

#### rsync命令格式常见的三种：

1.rsync [OPTION]... SRC DEST

2.rsync [OPTION]... SRC [USER@]HOST:DEST rsync

例：rsync -avz --delete /SRC -e ssh root@172.16.12.129:/DEST

需要修改端口时：

rsync -avz --delete /SRC -e "ssh -p2222" root@172.16.12.129:/DEST

3.[OPTION]... [USER@]HOST:SRC DEST

#### rsync的参数说明

-v：详细输出

-z：传输时进行压缩以提高传输效率。

-a：归档模式，表示以递归的方式传输文件，并保持文件的属性

--exclude：排除不需要同步传输的文件或者目录  
--delete：让目标目录和源目录的数据一致

## Inotify介绍：

Inotify是一种强大的、细粒度的、异步的文件系统事件监控机制，linux内核从2.6.13起，加入了Inotify支持，通过Inotify可以监控文件系统中添加、删除，修改、移动等各种细微事件，利用这个内核接口，第三方软件就可以监控文件系统下文件的各种变化情况，而inotify-tools就是这样的一个第三方软件。  
在上面章节中，我们讲到，rsync可以实现触发式的文件同步，但是通过crontab守护进程方式进行触发，同步的数据和实际数据会有差异，而inotify可以监控文件系统的各种变化，当文件有任何变动时，就触发rsync同步，这样刚好解决了同步数据的实时性问题。

## 环境说明

服务类型	IP地址	应用	操作系统
源服务器	192.168.24.248	rsync inotify-tools 脚本	centos7/redhat7
目标服务器	192.168.24.129	rsync	centos7/redhat7

## 需求

\*部署rsync+inotify-tools把源服务器上的etc目录同步到目标服务器上的/linfan/目录下

### 在目标服务器上做以下配置

#### 1.关闭防火墙与SELINUX

```
[root@linfan ~]# systemctl stop firewalld
[root@linfan ~]# systemctl disable firewalld
[root@linfan ~]# sed -ri 's/^(SELINUX=).*?/1disabled/g' /etc/sysconfig/selinux
[root@linfan ~]# getenforce 0
```

#### 2.安装rsync服务端软件

```
[root@linfan ~]# yum -y install rsync
```

#### 3.设置rsyncd.conf配置文件

```
[root@linfan ~]# vim /etc/rsyncd.conf //增加以下内容
log file = /var/log/rsyncd.log //日志文件位置，启动rsync后自动产生，无需提前创建
pidfile = /var/run/rsyncd.pid //pid文件存放位置
lock file = /var/run/rsync.lock //支持max connections参数的锁文件
secrets file = /etc/rsync.pass //用户认证配置文件,里面存放用户名和密码，必须手动创建这个文件

[etc_from_client] //自定义同步名称
path = /linfan/ //rsync服务端存放路径，客户端的数据将同步到此目录
comment = sync etc from client
uid = root //设置rsync运行权限为root
gid = root //设置rsync运行权限为root
port = 873 //默认端口为873
ignore errors //表示出现错误忽视错误
use chroot = no //默认为true，修改为no，增加对目录软链接的备份
read only = no //设置rsync服务端为读写权限
list = no //不显示rsync服务端资源列表
max connections = 200 //最大连接数
timeout = 600 //设置超时时间
auth users = admin //执行数据同步的用户名，可以设置多个，用英文逗号隔开
hosts allow = 192.168.24.248 //允许进行数据同步的IP地址，可以设置多个，用英文逗号隔开
```

```
hosts deny = 192.168.24.188 ///禁止进行数据同步的IP地址，可以设置多个，用英文逗号隔开
```

#### 4.创建存放路径目录

```
[root@linfan ~]# mkdir /linfan
```

#### 5.创建用户认证文件

```
[root@linfan ~]# echo 'admin:518' > /etc/rsync.pass
[root@linfan ~]# cat /etc/rsync.pass
admin:518
```

#### 6.设置文件权限

```
[root@linfan ~]# chmod 600 /etc/rsync*
[root@linfan ~]# ll /etc/rsync*
-rw-----. 1 root root 880 Aug 13 14:54 /etc/rsyncd.conf
-rw-----. 1 root root 10 Aug 13 14:55 /etc/rsync.pass
```

#### 7.启动rsync服务并设置开机自启动

```
[root@linfan ~]# systemctl start rsyncd
oot@linfan ~]# systemctl enable rsyncd
Created symlink from /etc/systemd/system/multi-user.target.wants/rsyncd.service to /usr/
```

在源服务器上做以下部署：

##### 1.关闭防火墙与SELINUX

```
[root@linfan ~]# systemctl stop firewalld
[root@linfan ~]# systemctl disable firewalld
[root@linfan ~]# sed -ri 's/^(SELINUX=).*1disabled/g' /etc/sysconfig/selinux
[root@linfan ~]# setenforce 0
```

##### 2.配置yum源

```
[root@linfan ~]# cd /etc/yum.repos.d/
[root@linfan yum.repos.d]# wget http://mirrors.163.com/help/CentOS7-Base-163.repo
[root@linfan yum.repos.d]# sed -i 's/|$releasever/7/g' /etc/yum.repos.d/CentOS7-Base-163.repo
[root@linfan yum.repos.d]# sed -i 's/^enabled=.*enabled=1/g' /etc/yum.repos.d/CentOS7-Base-163.repo
[root@linfan ~]# yum -y install epel-release //安装过程略
[root@linfan ~]# yum -y update --skip-broken //升级过程略
```

安装rsync服务端软件，只需要安装，不要启动，不需要配置

```
[root@linfan ~]# yum -y install rsync
```

##### 3.创建认证密码文件

```
[root@linfan ~]# echo '518' > /etc/rsync.pass
[root@linfan ~]# cat /etc/rsync.pass
518
```

#### 4. 设置文件权限，只设置文件所有者具有读取、写入的权限

```
[root@linfan ~]# chmod 600 /etc/rsync.pass
[root@linfan ~]# ll /etc/rsync.pass
-rw-----. 1 root root 0 Aug 13 15:46 /etc/rsync.pass
```

#### 5. 在源服务器上创建测试目录，然后在源服务器上运行以下命令

```
[root@linfan ~]# mkdir -pv /root/etc/test
rsync -avH --port 873 --progress --delete /root/etc/ admin@192.168.24.129::etc_from_client --p
sending incremental file list
./
test/

sent 69 bytes received 22 bytes 182.00 bytes/sec
total size is 0 speedup is 0.00
```

#### 6. 运行完成后在目标服务器上查看，在/linfan/目录下有test目录，说明数据同步成功

```
[root@linfan ~]# ls /linfan
test
```

#### 7. 安装inotify-tools工具，实时触发rsync同步

```
//检查服务器内核是否支持inotify
[root@linfan ~]# ll /proc/sys/fs/inotify/
total 0
-rw-r--r--. 1 root root 0 Aug 13 16:13 max_queued_events
-rw-r--r--. 1 root root 0 Aug 13 16:13 max_user_instances
-rw-r--r--. 1 root root 0 Aug 13 16:13 max_user_watches
//如果有这三个max开头的文件则表示服务器内核支持inotify

//安装inotify-tools
[root@linfan ~]# yum -y install make gcc gcc-c++ inotify-tools
```

#### 8. 写同步脚本

```
[root@linfan ~]# mkdir /scripts
[root@linfan ~]# touch /scripts/inotify.sh
[root@linfan ~]# chmod 755 /scripts/inotify.sh
[root@linfan ~]# ll
total 8
drwxr-xr-x. 3 root root 18 Aug 13 15:48 ad
-rw-----. 1 root root 1309 Jul 11 15:50 anaconda-ks.cfg
-rw-r--r--. 1 root root 71 Jul 18 17:16 doudou.repo
drwxr-xr-x. 3 root root 18 Aug 13 15:48 etc
[root@linfan ~]# vim /scripts/inotify.sh
host=192.168.24.129 //目标服务器的ip (备份服务器)
src=/etc //在源服务器上所要监控的备份目标
des=etc_from_client //自定义的模块名，需要与目标服务器上的定义名称同步
password=/etc/rsync.pass //执行数据同步的密码文件
user=admin //执行数据同步的名
inotifywait=/usr/bin/inotifywait

$inotifywait -mrq --timefmt '%Y%m%d%H:%M' --format '%T %w%f%e' -e modify,delete,create,c
| while read files ; do
    rsync -avzP --delete --timeout=100 --password-file=${password} $src $user@$host::$de
    echo "${files} was rsynced" >>/tmp/rsync.log 2>&1
done
```

//检查脚本

```
[root@linfan ~]# bash -x /scripts/inotify.sh
```

//启动脚本

```
[root@linfan ~]# nohup bash /scripts/inotify.sh &
[root@linfan ~]# ps -ef|grep inotify
```

//在源服务器上生成一个新的文件

```
[root@linfan ~]# mkdir /etc/httpd24/
[root@linfan ~]# echo 'hello world' > /etc/httpd24/test
```

//查看inotify生成的日志

```
[root@linfan ~]# tail /tmp/rsync.log
20180810 14:59 /etc/httpd24/testCREATE was rsynced
20180810 14:59 /etc/httpd24/testMODIFY was rsynced
//从日志上可以看到，生成了一个test文件，并且添加内容到其中
```

## 9.设置脚本开机自动启动

```
[root@linfan ~]# chmod +x /etc/rc.d/rc.local
[root@linfan ~]# ll /etc/rc.d/rc.local
-rwxr-xr-x. 1 root root 473 Apr 11 15:36 /etc/rc.d/rc.local
[root@linfan ~]# echo 'nohup /bin/bash /scripts/inotify.sh' >> /etc/rc.d/rc.local
[root@linfan ~]# tail /etc/rc.d/rc.local
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
```

## 10.到目标服务器上查看是否把新生成的文件自动传上去了

```
[root@linfan linfan]# pwd
/linfan
[root@linfan linfan]# ls
etc test
[root@linfan ~]# cd /etc/httpd24/
[root@linfan http24]# cat test
hello world
//在源服务器上的内容已经同步目录上了
```

©著作权归作者所有：来自51CTO博客作者上古杰作的原创作品，如需转载，请注明出处，否则将追究法律责任

rsync      Notify

服务管理

0

收藏      分享

上一篇：[httpd/apache编译安装...](#)      下一篇：[mysql基础解析](#)

**上古杰作**

49篇文章, 30W+人气, 28粉丝  
以初学者的心态学习知识

[关注](#)

提问和评论都可以, 用心的回复会被更多人看到和认可

Ctrl+Enter 发布

取消

[发布](#)

推荐专栏

[更多](#)

### 带你玩转高可用

前百度高级工程师的架构高可用实战  
共15章 | 曹林华

¥ 51.00    506人订阅

[订 阅](#)

### 负载均衡高手炼成记

高并发架构之路  
共15章 | sery

¥ 51.00    573人订阅

[订 阅](#)

### 基于Python的DevOps实战

自动化运维开发新概念  
共20章 | 抚琴煮酒

¥ 51.00    515人订阅

[订 阅](#)

### 网工2.0晋级攻略——零基础入门Python/A...

网络工程师2.0进阶指南  
共30章 | 姜汁啤酒

¥ 51.00    1948人订阅

[订 阅](#)

### 全局视角看大型园区网

路由交换+安全+无线+优化+运维  
共40章 | 51CTOsummer

¥ 51.00    2367人订阅

[订 阅](#)

猜你喜欢

[带你见证expect的强大——全程自动化执行命令](#)[rsync详解之exclude排除文件](#)[rsync 安装使用详解](#)[调用ZABBIX-API进行大批量新增监控主机, 解放运维监...](#)[Java:使用wait\(\)与notify\(\)实现线程间协作](#)[keepalived中vrrp\\_script,track\\_script,notify的使用方法](#)

[sersync+rsync原理及部署](#)

[rsync传输性能测试总结](#)

[rsync故障排除解答](#)

[kubernetes 部署Prometheus](#)

[docker容器优雅停机导致的问题说明](#)

[通过docker-compose制作dubbo-admin和zookeeper...](#)

[Ubuntu如何使用root用户登录图形化界面？](#)

[rsync参数详解、利用ssh、rsync 实现数据的定时同步](#)

[rsync常见问题及解决办法收集](#)

[linux rsync 指定用户名和密码的方式同步](#)

[VMWARE下复制centos8虚拟机导致IP丢失问题处理](#)

[kubernetes 的安全](#)

[kubernetes 的调度](#)

[CentOS 7.7 rsync使用详解，并结合inotify实现文件实...](#)