# rsync+inotify主从同步方案

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 撰稿人 | 日期 | 说明 |
| V1.0 | 张凯 | 2016年9月27日 | 保证集群内的主机项目一致性，并可实现自动更新 |
|  |  |  |  |

目录

[rsync+inotify主从同步方案 1](#_Toc27114)

[一、应用场景说明 2](#_Toc4253)

[二、rsync 和Inotify 简单介绍 2](#_Toc5022)

[三、Rsync 和Inotify安装及脚本 3](#_Toc18810)

[四、 配置rsync server 4](#_Toc16059)

[五、启动rsync server 6](#_Toc24617)

[六、Inotify 监控脚本 7](#_Toc26433)

[七、Inotify 脚本 7](#_Toc769)

# 一、应用场景说明

考虑到目前生产集群内主机的配置及项目其实都一样，可以考虑做成同步，即集群内主机只全部与第一台保持同步，这样做的好处是以后更新只需要更新集群中的一台主机即可，其余自动更新。

# 二、rsync 和Inotify 简单介绍

1） rsync是类unix系统下的数据镜像备份工具,rsync命令是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件。rsync使用所谓的“rsync算法”来使本地和远程两个主机之间的文件达到同步，这个算法只传送两个文件的不同部分，而不是每次都整份传送，文件传输效率高，很容易做到保持原来文件的权限、时间、软硬链接等等。

2）Inotify可以监控文件系统操作，比如读取、写入和创建等，通过和rsync配合可以做到将数据实时的进行单向同步。

# 三、Rsync 和Inotify安装及脚本

1）Rsync安装：

#yum install xinetd -y

#yum install rsync -y

2）Inotify 安装

#wget http://tenet.dl.sourceforge.net/project/inotify-tools/inotify-tools/3.13/inotify-tools-3.13.tar.gz

#cd inotify-tools-3.13

# ./configure

# make

# sudo make install

# 配置rsync server

1、xinetd配置及rsync启动

###在server1中执行.

#启动RSYNC  
#vi /etc/xinetd.d/rsync  
#把原来的YES改成NO  
service rsync  
{  
        disable = no  
        socket\_type     = stream  
        wait            = no  
        user            = root  
        server          = /usr/bin/rsync  
        server\_args     = --daemon  
        log\_on\_failure  += USERID  
}  
#随系统启动RSYNC

#chkconfig rsync on

2、配置/etc/rsyncd.conf(需要手动生成)

uid = root

gid = root

use chroot = no

max connections = 8

strict modes = yes

port = 873

[data]

path = /root/test

comment = welcome

ignore errors

read only = no

list = no

auth users = kaiz

secrets file = /etc/rsync.passwd

hosts allow = 172.29.19.100

hosts deny=0.0.0.0/0

#pid file = /var/run/rsyncd.pid

#lock file = /var/run/rsync.lock

#log file = /var/log/rsyncd.log

###在server2中执行.

1. 配置rsync密码（在上边的配置文件中已经写好路径） /etc/rsync.pas（名字随便写，只要和上边配置文件里的一致即可），格式(一行一个用户)

#vi /etc/rsync.passwd  
kaiz:123456

权限：因为rsync.pas存储了rsync服务的用户名和密码，所以非常重要。要将rsync.pas设置为root拥有, 且权限为600。

#cd /etc  
#chown root.root rsync.pas   
#chmod 600 rsync.pas

注：rsyncd.motd（配置欢迎信息，可有可无）

若有多个server可均按此配置

# 五、启动rsync server

**RSYNC服务端启动的两种方法**  
1、启动rsync服务端（独立启动）  
#/usr/bin/rsync –daemon

2、启动rsync服务端 （有xinetd超级进程启动）  
# /etc/rc.d/init.d/xinetd reload

3、加入rc.local   
在各种操作系统中，rc文件存放位置不尽相同，可以修改使系统启动时把rsync --daemon加载进去。  
#vi /etc/rc.local  
加入一行/usr/bin/rsync --daemon

# 六、Inotify 监控脚本

同步目录：/root/test/test1

# 七、Inotify 脚本

#!/bin/env /bash/sh

###AUTHOR:zhangkai

###DATE:2016.9.8

#VERSION: V1.1

#FUNCTION: To keep server and client same all times

src=/data/ # 需要同步的源路径

des=data # 目标服务器上 rsync --daemon 发布的名称，

rsync\_passwd\_file=/etc/rsync.passwd # rsync验证的密码文件

ip1=172.29.19.41 # 目标服务器1

ip2=172.29.19.45 # 目标服务器2

user=kaiz # rsync --daemon定义的验证用户名

/usr/local/bin/inotifywait -mrq --format '%Xe %w%f' -e modify,create,delete,attrib,close\_write,move $src | while read file #此处应为绝对路径，把监控到有发生更改的"文件路径列表"循环

do

INO\_EVENT=$(echo $file | awk '{print $1}') # 把inotify输出切割 把事件类型部分赋值给INO\_EVENT

INO\_FILE=$(echo $file | awk '{print $2}') # 把inotify输出切割 把文件路径部分赋值给INO\_FILE

echo "-------------------------------$(date)------------------------------------"

echo $file

#增加、修改、写入完成、移动进事件

#增、改放在同一个判断，因为他们都肯定是针对文件的操作，即使是新建目录，要同步的也只是一个空目录，不会影响速度。

if [[ $INO\_EVENT =~ 'CREATE' ]] || [[ $INO\_EVENT =~ 'MODIFY' ]] || [[ $INO\_EVENT =~ 'CLOSE\_WRITE' ]] || [[ $INO\_EVENT =~ 'MOVED\_TO' ]] # 判断事件类型

then

echo 'CREATE or MODIFY or CLOSE\_WRITE or MOVED\_TO'

rsync -avzcr --password-file=${rsync\_passwd\_file} $(dirname ${INO\_FILE})/ ${user}@${ip1}::${des} && rsync -avzcr --password-file=${rsync\_passwd\_file} $(dirname ${INO\_FILE})/ ${user}@${ip2}::${des}

fi

#删除、移动出事件

if [[ $INO\_EVENT =~ 'DELETE' ]] || [[ $INO\_EVENT =~ 'MOVED\_FROM' ]]

then

echo 'DELETE or MOVED\_FROM'

rsync -avzr --delete --password-file=${rsync\_passwd\_file} $(dirname ${INO\_FILE})/ ${user}@${ip1}::${des} &&

rsync -avzr --delete --password-file=${rsync\_passwd\_file} $(dirname ${INO\_FILE})/ ${user}@${ip2}::${des}

fi

if [[ $INO\_EVENT =~ 'ATTRIB' ]]

then

echo 'ATTRIB'

if [ ! -d "$INO\_FILE" ]

then

rsync -avzcr --password-file=${rsync\_passwd\_file} $(dirname ${INO\_FILE})/ ${user}@${ip1}::${des} &&

rsync -avzcr --password-file=${rsync\_passwd\_file} $(dirname ${INO\_FILE})/ ${user}@${ip2}::${des}

fi

fi

done