

## sersync 同步配置

### 同步目标服务器配置

#### rsyncd daemon 服务配置（目标服务器）

```
vim /etc/rsyncd.conf
# Minimal configuration file for rsync daemon
# See rsync(1) and rsyncd.conf(5) man pages for help

# This line is required by the /etc/init.d/rsyncd script
# GLOBAL OPTIONS
uid = root
gid = root

use chroot = no
read only = false
# limit access to private LANs
#hosts allow = 192.168.1.0/24
hosts allow = 192.168.1.213, 192.168.1.214, 192.168.1.215
hosts deny = *
ignore errors
max connections = 2000
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
# This will give you a separate log file
log file = /var/log/rsync.log
auth users = rsync_backup
secrets file = /etc/rsyncd.password
# This will log every file transferred - up to 85,000+ per user, per sync
transfer logging = yes
log format = %t %a %m %f %b
syslog facility = local3
timeout = 300

# MODULE OPTIONS
[lmfile]
comment = sync lm static file
path = /alidata/static/lm/file
```

#### 创建rsync 的同步密码文件

```
# 写入密码文件
vim /etc/rsyncd.password
rsync_backup:fgJvz2vxRpeld3b

# 密码文件的权限必须为 600
chmod 600 /etc/rsyncd.password
```

#### 启动守护进程，配置自启

```
# 启动服务
/usr/bin/rsync --daemon

# 配置自启
vim /etc/rc.local
/usr/bin/rsync --daemon
```

## 进行同步的测试

```
# 在 同步源 服务器上配置
# 建立密码文件
vim /etc/rsync.password
fgJvz2vxRpeld3b

# 密码文件的权限必须为 600
chmod 600 /etc/rsync.password

# 进行同步的测试
rsync -av /alidata/data/mongo/ rsync_backup@192.168.1.212::lmfile/ --password-
file=/etc/rsync.password
或
rsync -av /alidata/data/mongo/ rsync://rsync_backup@192.168.1.212/lmfile/ --password-
file=/etc/rsync.password
```

## 同步源服务器配置

```
# 下载 sersync
cd /usr/local/src
wget https://sersync.googlecode.com/files/sersync2.5.4_64bit_binary_stable_final.tar.gz

mv GNU-Linux-x86 /usr/local/sersync
mkdir bin conf logs
mv confxml.xml conf && mv sersync2 bin/
```

## 更改confxml.xml 配置文件

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<head version="2.5">
  <host hostip="localhost" port="8008"></host>
  <debug start="false"/>
  <fileSystem xfs="false"/>
  <filter start="false">
    <exclude expression="(.)\.svn"></exclude>
    <exclude expression="(.)\.gz"></exclude>
    <exclude expression="^info/*"></exclude>
    <exclude expression="^static/*"></exclude>
  </filter>
  <inotify>
    <delete start="true"/>
    <createFolder start="true"/>
```

```

<createFile start="false"/>
<closeWrite start="true"/>
<moveFrom start="true"/>
<moveTo start="true"/>
<attrib start="false"/>
<modify start="false"/>
</inotify>

<sersync>
  <!--
<localpath watch="/alidata/static/lm/file/">
  -->
<localpath watch="/alidata/data/mongo/">
  <remote ip="192.168.1.212" name="lmfile"/>
  <!--<remote ip="192.168.8.39" name="tongbu"/>-->
  <!--<remote ip="192.168.8.40" name="tongbu"/>-->
</localpath>
<rsync>
  <commonParams params="-artuz"/>
  <auth start="true" users="rsync_backup" passwordfile="/etc/rsync.password"/>
  <userDefinedPort start="false" port="874"/><!-- port=874 -->
  <timeout start="true" time="100"/><!-- timeout=100 -->
  <ssh start="false"/>
</rsync>
<failLog path="/usr/local/sersync/logs/rsync_fail_log.sh" timeToExecute="60"/><!--default
every 60mins execute once-->
<crontab start="false" schedule="600"/><!--600mins-->
  <crontabfilter start="false">
    <exclude expression="*.php"></exclude>
    <exclude expression="info/*"></exclude>
  </crontabfilter>
</crontab>
<plugin start="false" name="command"/>
</sersync>

<plugin name="command">
  <param prefix="/bin/sh" suffix="" ignoreError="true"/> <!--prefix /opt/tongbu/mmm.sh
suffix-->
  <filter start="false">
    <include expression="(.)\.php"/>
    <include expression="(.)\.sh"/>
  </filter>
</plugin>

<plugin name="socket">
<localpath watch="/opt/tongbu">
  <deshost ip="192.168.138.20" port="8009"/>
</localpath>
</plugin>
<plugin name="refreshCDN">
<localpath watch="/data0/htdocs/cms.xoyo.com/site/">
  <cdninfo domainname="ccms.chinacache.com" port="80" username="xxxx" passwd="xxxx"/>

  <sendurl base="http://pic.xoyo.com/cms"/>

```

```

        <regexurl regex="false" match="cms.xoyo.com/site([a-zA-Z0-9]*).xoyo.com/images"/>
    </localpath>
</plugin>
</head>

```

## 配置 环境变量

```

echo 'export PATH=$PATH:/usr/local/sersync/bin' >> /etc/profile
. /etc/profile

```

## 启动 sersync

```

[root@Tidb-1 conf]# sersync2 -r -d -o /usr/local/sersync/conf/confxml.xml
set the system param
execute: echo 50000000 > /proc/sys/fs/inotify/max_user_watches
execute: echo 327679 > /proc/sys/fs/inotify/max_queued_events
parse the command param
option: -r  rsync all the local files to the remote servers before the sersync work
option: -d  run as a daemon
option: -o  config xml name:  /usr/local/sersync/conf/confxml.xml
daemon thread num: 10
parse xml config file
host ip : localhost host port: 8008
daemon start, sersync run behind the console
use rsync password-file :
user is rsync_backup
passwordfile is  /etc/rsync.password
config xml parse success
please set /etc/rsyncd.conf max connections=0 Manually
sersync working thread 12 = 1(primary thread) + 1(fail retry thread) + 10(daemon sub threads)
Max threads numbers is: 22 = 12(Thread pool nums) + 10(Sub threads)
please according your cpu , use -n param to adjust the cpu rate
-----
rsync the directory recursivly to the remote servers once
working please wait...
execute command: cd /alidata/data/mongo && rsync -artuz -R --delete ./ --timeout=100
rsync_backup@192.168.1.212::lmfile --password-file=/etc/rsync.password >/dev/null 2>&1
[root@Tidb-1 conf]# run the sersync:
watch path is: /alidata/data/mongo

```

## 默认的执行的命令为

```

execute command: cd /alidata/data/mongo && rsync -artuz -R --delete ./ --timeout=100
rsync_backup@192.168.1.212::lmfile --password-file=/etc/rsync.password >/dev/null 2>&1

```

存在有 --delete 参数，不适用多个的服务器目录，同步到同一个的目录。

sersync2 -r -d -o /usr/local/sersync/conf/confxml.xml (指定 -r 之后，会进行强一致的同步)

# 首次最好，启动服务时使用

```

sersync2 -d -o /usr/local/sersync/conf/confxml.xml

```

进行启动，避免进行强一致的同步，导致不能两台服务的文件同步到一台服务器的同一目录

## Lsyncd 同步配置

### Lsyncd 的安装 (CentOS)

```
# 项目地址进行下载 zip 包
https://github.com/axkibe/lsyncd

# 安装依赖
yum install lua lua-devel

# 进行编译安装
unzip lsyncd-master.zip
cd lsyncd-master
cmake -DCMAKE_INSTALL_PREFIX=/usr/local/lsyncd
make && make install

# 创建配置文件
cd /usr/local/lsyncd/
mkdir ./conf
mkdir ./var
vim conf/lsyncd.conf
```

### 配置文件

```
-- 全局配置
settings {
    -- 日志文件存放位置
    logfile = "/usr/local/lsyncd/logs/lsyncd.log",

    -- 状态文件存放位置
    statusFile = "/usr/local/lsyncd/var/lsyncd.status",

    -- 将lsyncd的状态写入上面的statusFile的间隔，默认10秒
    --statusInterval = 10

    -- 是否启用守护模式，默认 true
    --nodaemon=true

    -- inotify监控的事件，默认是 CloseWrite,还可以是 Modify 或 CloseWrite or Modify
    inotifyMode = "CloseWrite",

    -- 最大同步进程
    maxProcesses = 8,

    --累计到多少所监控的事件激活一次同步，即使后面的delay延迟时间还未到，打开此选项实时进行同步
    maxDelays = 1
}

-- 远程目录同步
sync {
```

```

-- rsync , rsyncssh , direct 三种模式
default.rsync,

-- 同步的源目录, 使用绝对路径。
source = "/alidata/static/lm/file/",

-- 定义目的地址.对应不同的模式有几种写法,这里使用远程同步的地址, rsync中的地址
target = "rsync_backup@192.168.1.212::lmfile/",

-- 默认 true , 允许同步删除。还有 false, startup, running 值, true 会保持强一致
delete = false,

-- 哪些文件不同步, 以 点 开头的不进行同步
exclude = { ".*" },

-- 累计事件, 等待rsync同步延时时间, 默认15秒,最大累计到1000个不可合并的事件(1000个文件变动),
delay = 5,

-- 默认 true 当init = false ,只同步进程启动以后发生改动事件的文件, 原有的目录即使有差异也不会同步
init = false,

-- rsync 的配置
rsync = {
    -- rsync 的二进制处理文件
    binary = "/usr/bin/rsync",

-- 归档模式
    archive = true,

-- 压缩传输
    compress = true,

-- 增量
    verbose = true,

-- 密码文件
    password_file = "/etc/rsync.password",

-- 其他 rsync 的配置参数, 限速(--bwlimit Kbps),使用 rsync -v 查看详细参数
    -- _extra = {"--bwlimit=200"}
}
}

```

## 配置rsync.password 密码文件

```

cat /etc/rsync.password
fgJvz2vxRpeld3b

```

## 加入环境变量

```

ln -s /usr/local/lsyncd/bin/lsyncd /usr/bin/lsyncd

```

## 启动Lsyncd

```
lsyncd -log Exec /usr/local/lsyncd/conf/lsyncd.conf
```

## Lsyncd 其他的配置项说明

settings

里面是全局设置，--开头表示注释，下面是几个常用选项说明：

logfile 定义日志文件

stausFile 定义状态文件

nodaemon=true 表示不启用守护模式，默认

statusInterval 将lsyncd的状态写入上面的statusFile的间隔，默认10秒

inotifyMode 指定inotify监控的事件，默认是CloseWrite，还可以是Modify或CloseWrite or Modify

maxProcesses 同步进程的最大个数。假如同时有20个文件需要同步，而maxProcesses = 8，则最大能看到有8个

rsync进程

maxDelays 累计到多少所监控的事件激活一次同步，即使后面的delay延迟时间还未到

sync

里面是定义同步参数，可以继续使用maxDelays来重写settings的全局变量。一般第一个参数指定lsyncd以什么模式运行：rsync、rsyncssh、direct三种模式：

default.rsync：本地目录间同步，使用rsync，也可以达到使用ssh形式的远程rsync效果，或daemon方式连接远程rsyncd进程；

default.direct：本地目录间同步，使用cp、rm等命令完成差异文件备份；

default.rsyncssh：同步到远程主机目录，rsync的ssh模式，需要使用key来认证

source 同步的源目录，使用绝对路径。

target 定义目的地址.对应不同的模式有几种写法：

/tmp/dest：本地目录同步，可用于direct和rsync模式

172.29.88.223:/tmp/dest：同步到远程服务器目录，可用于rsync和rsyncssh模式，拼接的命令类似

于/usr/bin/rsync -ltsd --delete --include-from=- --exclude=\* SOURCE TARGET，剩下的就是rsync的内容了，比如指定username，免密码同步

172.29.88.223::module：同步到远程服务器目录，用于rsync模式

三种模式的示例会在后面给出。

init 这是一个优化选项，当init = false，只同步进程启动以后发生改动事件的文件，原有的目录即使有差异也不会同步。默认是true

delay 累计事件，等待rsync同步延时时间，默认15秒（最大累计到1000个不可合并的事件）。也就是15s内监控目录下发生的改动，会累积到一次rsync同步，避免过于频繁的同步。（可合并的意思是，15s内两次修改了同一文件，最后只同步最新的文件）

excludeFrom 排除选项，后面指定排除的列表文件，如excludeFrom = "/etc/lsyncd.exclude"，如果是简单的排除，可以使用exclude = LIST。

这里的排除规则写法与原生rsync有点不同，更为简单：

监控路径里的任何部分匹配到一个文本，都会被排除，例如/bin/foo/bar可以匹配规则foo

如果规则以斜线/开头，则从头开始要匹配全部

如果规则以/结尾，则要匹配监控路径的末尾

?匹配任何字符，但不包括/

\*匹配0或多个字符，但不包括/

**\*\*匹配0或多个字符，可以是/**

**delete** 为了保持target与source完全同步，Lsyncd默认会delete = true来允许同步删除。它除了false，还有startup、running值，请参考 Lsyncd 2.1.x || Layer 4 Config || Default Behavior。

**rsync**

(提示一下，delete和exclude本来都是rsync的选项，上面是配置在sync中的，我想这样做的原因是为了减少rsync的开销)

**bwlimit** 限速，单位kb/s，与rsync相同 (这么重要的选项在文档里竟然没有标出)

**compress** 压缩传输默认为true。在带宽与cpu负载之间权衡，本地目录同步可以考虑把它设为false

**perms** 默认保留文件权限。

其它rsync的选项

其它还有rsyncssh模式独有的配置项，如host、targetdir、rsync\_path、password\_file，见后文示例。

rsyncOps={"-avz","--delete"}这样的写法在2.1.\*版本已经不支持。

lsyncd.conf可以有多个sync，各自的source，各自的target，各自的模式，互不影响。

## Ubuntu 安装Lsyncd

```
apt install lsyncd
```

```
# 创建配置文件目录
```

```
cd /usr/local/lsyncd/
```

```
mkdir ./conf
```

```
mkdir ./var
```

```
vim conf/lsyncd.conf
```

```
# 连接二进制文件
```

```
cd /usr/local/lsyncd/bin
```

```
ln -s /usr/bin/lsyncd lsyncd
```

```
# 更改默认的系统的启动的脚本
```

```
vim /etc/init.d/lsyncd
```

```
PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/lsyncd/bin    # 添加路径/usr/local/lsyncd/bin
```

```
DESC="synchronization daemon"
```

```
NAME=lsyncd
```

```
DAEMON=/usr/local/lsyncd/bin/$NAME    # 指定lsyncd 的路径
```

```
CONFIG=/usr/local/lsyncd/conf/lsyncd.conf    # 指定配置文件的路径
```

```
PIDFILE=/var/run/$NAME.pid
```

```
DAEMON_ARGS="-pidfile ${PIDFILE} ${CONFIG}"
```

```
SCRIPTNAME=/etc/init.d/$NAME
```

```
NICELEVEL=10
```

```
# ubuntu 启动 lsyncd
```

```
/etc/init.d/lsyncd start
```