

# restic使用rclone后端备份数据至WebDAV服务器

Published on Oct 30, 2022 UTC by Wang Ye <<https://wangye.org>>

备份是运维避不开的一个话题，架构再精巧的服务器环境或者关键业务数据总会遇到不可控的事件而在顷刻之间遭到摧毁。勒索病毒、黑客入侵、主机商跑路、员工删库等等，这些不可控因素仿佛是悬在运维头顶的达摩克利斯之剑，一旦发生就要靠最后时刻的备份救场了，正所谓后方有粮，心中不慌。

手动的脱机备份也是正常运维的一部分，一般情况下我在每次大的修改配置或者升级系统前会将数据脱机备份一份，以备在发生问题时可以迅速恢复。对于云主机可以设置快照（Snapshots），或者直接cp一份，但这种简单的手动备份不仅耗时耗力而且容易遗忘，大部分情况下我们可能就仅仅是本地备份，无法做到异地灾备，所以我们就不得不考虑软件远程备份作为兜底。

本文就介绍了我在实际运维中所采用的几种远程备份方式供读者参考，最后介绍的restic是我比较推荐的一种备份方式。

## 1 常见的备份途径

工作室里有一台24小时运行的网络存储器（NAS），数据存储 NAS 上应该是最为稳妥的，所以我希望远程的数据能够在设定好的时间自动备份到工作室的网络存储器上。

### 1.1 群晖（Synology）Cloud Backup

对于群晖这种商业一站式解决方案，其必然会包含有备份工具，例如Synology Drive Cloud Backup系列，通过安装其专用的APP可自动备份对应数据，一般我的Windows主机使用的就是这套工具，遗憾的是虽然群晖官网上提供了全平台的APP客户端，但是其Linux客户端是GUI版本，且无开源参考，也就不太适用于纯命令行的服务器系统上了，所以对于Linux服务器的数据备份，这条路行不通。

实际上群晖除了使用自带的商业备份软件外，还支持rsync、FTP、WebDAV等形式的文件传输协议或者渠道，利用好它们依然可以实现我们的备份需求。

## 1.2 rsync命令进行远程备份

rsync应该是Linux系统上用的较多的同步软件，通过rsync可以便捷安全的将源设备上的文件备份到目标设备上，备份的通道依托于SSH，所以对于rsync备份来说，我们需要先配置好SSH连接，强烈建议在目标设备上建立普通身份的用户，仅限制权限到指定备份文件夹读写，然后只通过SSH的公私钥认证一种方式登录。

这里参考Joel Hans的[《Back Up Your Files With rsync And A VPS》](#)举一个例子：

```
#!/bin/sh

# ref: https://blog.ssdnodes.com/blog/back-up-rsync-vps/

# Create a timestamp
date=`date "+%Y-%m-%dT%H_%M_%S" `

# Specify the folder to snapshot on your local machine
SOURCE=/var/lib/docker/volumes/

# Specify the destination folder on your VPS
DEST=/home/wangye/snapshots/wangye.org

BACKUP_SERVER_HOST=mybackup.example.com
BACKUP_SERVER_PORT=22
BACKUP_SERVER_USER=wangye

# Execute rsync followed by cleanup
rsync -azvP4 \
  --delete \
  --link-dest=../current \
  -e 'ssh -p 22' \
  $SOURCE $BACKUP_SERVER_USER@$BACKUP_SERVER_HOST:$DEST/backup-$date \
```

```
&& ssh -4 $BACKUP_SERVER_USER@$BACKUP_SERVER_HOST -p  
$BACKUP_SERVER_PORT \  
"rm -rf $DEST/current \  
&& ln -s $DEST/backup-$(date +%Y-%m-%d) $DEST/current"
```

上述代码是根据本站实际所改写的备份代码，主要是将SOURCE变量所指示的路径/var/lib/docker/volumes/下内容备份至目的主机mybackup.example.com（由BACKUP\_SERVER\_HOST变量所指示）的/home/wangye/snapshots/wangye.org路径（由DEST变量所指示）之下。

同时备份由backup-打头，当前日期结尾，并自动设置备份主机上的current软链接指向最新的备份，更多的细节可以参考Joel Hans的原文，这里不再赘述。

当然需要注意的是这里推荐的是公私钥登录SSH方式，如果你知道风险仍然需要使用账号密码方式登录，那就需要借助于sshpass这个小工具了。

## 2 使用restic备份

到这里需要隆重介绍本文的主角restic，restic是一款开源免费而且跨平台的备份工具，这是本人对于服务器备份主要推荐的方式之一，使用restic的好处显而易见，比如加密传输、增量备份、多种备份手段，支持rclone后端等等。

下面列出restic支持的备份方式：

### 本机存储

```
sftp 服务器 (通过 SSH)  
HTTP REST 服务器 (protocol, rest-server)  
Amazon S3 (亦可使用支持此协议的Minio server)  
OpenStack Swift  
BackBlaze B2  
Microsoft Azure Blob Storage  
Google Cloud Storage  
其他受rclone支持的后端
```

## 2.1 restic与rclone的区别

在部署restic的时候，我发现原生支持的备份方式不适合对接到本地部署的群晖NAS，遂考虑使用rclone作为后端以增加可选择余地，其实rclone是一个支持大部分存储技术的同步软件，也可以单独使用rclone实现备份功能，但对于商业化备份来说，restic可能会更适合一些，这两种备份方式的区别比较大的主要有：

rclone是文件同步，也就是和大部分网盘软件一样，来源文件是什么样的，目的文件就是什么样的。restic是面向文件的专业备份工具，其在备份文件前将会先进行加密再传输，而且是增量备份，只备份变化的部分。

rclone并不记录文件版本，无法回溯特定时间点上的文件。restic会对每次备份生成一个快照，记录该时间点文件变化和存储结构，可以随时回溯到特定时间点的文件。

当然这两款优秀的工具还有其他区别，这里就不一一列出了，其实可以发现restic有点像项目开发所使用的版本控制软件，比如git或者svn，而rclone只是一个单纯的文件同步工具，有点像群晖的Cloud Drive。

## 2.2 部署restic备份数据至WebDAV

群晖NAS是支持WebDAV文件管理协议的，那么要实现将数据备份至群晖NAS就可以借助于WebDAV，而restic又并不能直接支持WebDAV，那么就需要借助rclone进行中转。

本着万物皆可docker的原则，对于restic的部署依然首选docker方式，这里选择[lobaro/restic-backup-docker](#)将代码库clone到本地src路径下，同时新建hooks和rclone目录，其中hooks用于存放备份开始前和开始后额外需要执行的命令，可以根据需要选择，rclone文件夹存放rclone的配置文件。

其中编写适用于群晖WebDAV的rclone配置文件rclone.conf如下（仅供参考，根据实际情况修改）：

```
[synology]
type = webdav
url = https://mybackup.example.com:8000
vendor = other
user = backupuser
pass = yourpassword
```

其中url指示WebDAV服务器地址和端口号，user和pass分别指示认证账号和密码。

在根路径下，编写docker-compose.yaml文件：

```
version: "3"
services:

  restic:
    container_name: restic
    build:
      context: ./src
      dockerfile: Dockerfile
    privileged: true
    restart: always
    environment:
      - TZ=Asia/Shanghai
      - 'RESTIC_REPOSITORY=rclone:synology:mybackup'
      - RESTIC_PASSWORD=ABC123
      - BACKUP_CRON=0 3 * * *
      - RESTIC_FORGET_ARGS=--keep-last 10
    dns:
      - 1.1.1.1
    volumes:
      - ./hooks:/hooks:ro
      - ./rclone:/root/.config/rclone:rw
      - /var/lib/docker/volumes:/data/dockervolumes:ro
      - /var/opt/backups:/data/backups:ro
```

为了确保备份容器能够访问服务器上的文件，这里赋予了`privileged: true`权限，如果你需要备份的文件不需要`root`权限也可以访问，那此配置项可以省略。

另外需要注意的是此`docker`部署默认会备份容器`/data`路径下的数据，所以我们可以将需要备份的内容挂载在`/data`路径下，这里我们挂载了服务器上`/var/lib/docker/volumes`和`/var/opt/backups`，表示需要备份此两处的文件，读者可以根据需要进行修改。

环境变量`RESTIC_REPOSITORY=rclone:synology:mybackup`指示`restic`使用`rclone`作为备份后端，其中`synology`表示使用`rclone.conf`的`synology`配置节，`mybackup`表示备份至群晖NAS的`mybackup`路径下，变量`BACKUP_CRON=0 3 * * *`指示备份开始的计划任务时间，这里设置为每天晚上3点开始，更多环境变量的含义和配置项可以参考[lobaro/restic-backup-docker](https://github.com/lobaro/restic-backup-docker)项目主页的描述。

使用`docker-compose up -d`拉起容器如果没有错误的话那说明配置成功完成。

### 3 总结

对于任何线上项目，数据永远是灵魂，如果不重视数据备份，待到出现误删除、VPS服务商跑路、勒索病毒攻击等重大灾难的时候将会让我们付出惨痛的代价，本文介绍了一种较易部署和简单可靠的备份方式，读者可以在实际运用中灵活变通以实现强大的数据备份功能。

## External References (Links)

《Back Up Your Files With rsync And A VPS》  
<https://blog.ssdnodes.com/blog/back-up-rsync-vps/>  
`restic`  
<https://restic.net/>

<https://wangye.org/posts/2022/10/backup-file-to-webdav-via-restic-and-rclone.html>

lobaro/restic-backup-docker

<https://github.com/Lobaro/restic-backup-docker>