

Скрипт резервного копирования через rsync

ОПУБЛИКОВАНО: 04.08.2014 | АВТОР: НИКОЛАЙ

Возникла необходимость как-то и куда-то бэкапиться. Причём чтобы процессоры не грузились и место не занималось, а бэкапы ротэйтились и удобно доставались. Раньше всегда пользовался [fsbackup](#), но захотелось отказаться от архивирования. Для решения задачи была использована rsync и механизм жёстких ссылок (так называемых хардлинков) файловой системы.

Архитектура: есть отдельно стоящий сервер с большим винтом — на нём и работает скрипт. Есть много разных серверов с доступом по ssh, на которых в ~/.ssh/authorized_keys добавлен публичный ключ пользователя, под которым работает скрипт резервного копирования.

Логика работы: в определённое время скрипт по ssh синхронизирует содержимое папки на удалённом сервере с папкой domain.com/latest, а потом копирует её в папку с сегодняшней датой, создавая при этом жёсткие ссылки на файлы, затем удаляет папки, дата создания которых старше 7 дней. Т.к. синхронизируется только содержимое каталога, дампить базу по крону нужно на клиентской машине перед тем, как rsync заберёт файлы.

Плюсы:

- использует меньше места, чем дифференциальные бэкапы и не больше места, чем инкрементальные
- меньше грузит процессор, т.к. не использует архиваторы (можно осуществлять сжатие на лету при передаче по сети)
- имеет достаточно подробный формата лога, оповещения по емэйлу об ошибках
- устойчив к взлому или полному уничтожению клиентской машины
- бэкапы злоумышленник не повредит никак

Вопрос:

— т.к. скрипт был первоначально опубликован в [бложике](#), так и не удалось услышать авторитетное мнение относительно эффективности такого подхода — был бы рад, если бы вы поделились мыслями...

```
#!/bin/sh
# simple rsync backup script written by farmal.in 2011-01-21
#
# latest backup is always in $SDIR/domains/$domain/latest folder
# all backups which are older than 7 days would be deleted
# backup.ini file can't contain comments, empty lines and spaces in
domain names
#
# example of a GOOD backup.ini:
# mydomain.com user@mydomain.com:/path/to/public_html
#

SDIR="/usr/local/backup"
SKEY="$SDIR/.ssh/id_rsa"
SLOG="$SDIR/backup.log"
PID_FILE="$SDIR/backup.pid"
ADMIN_EMAIL="email@domain.com"

if [ -e $PID_FILE ]; then
    echo "this task is already running or previous run was
completed with errors on `hostname`" | mail -s "Some mess with
backups on `hostname`..." $ADMIN_EMAIL
    exit
fi

touch $PID_FILE

# redirecting all output to logfile
exec >> $SLOG 2>&1

# parsing backup.ini file into $domain and $from variables
cat backup.ini | while read domain from ; do
    destination="$SDIR/domains/$domain"
    # downloading a fresh copy in 'latest' directory
    echo -e "`date` *** $domain backup started">>$SLOG

    # start counting rsync worktime
    start=$(date +%s)
    rsync --archive --one-file-system --delete -e "ssh -i $SKEY"
"$from" "$destination/latest" || (echo -e "Error when rsyncing
$domain. nn For more information see $SLOG:nn `tail $SLOG`" | mail
-s "rsync error" $ADMIN_EMAIL & continue)
    finish=$(date +%s)
    echo -e "`date` *** RSYNC worked for $((finish - start))
seconds">>$SLOG

    # cloning the fresh copy by hardlinking
    cp --archive --link "$destination/latest" "$destination/`date
+%F`"
    # deleting all previous copies which are older than 7 days by
creation date, but not 'latest'
    find "$destination" -maxdepth 1 -ctime +7 -type d -path
"$destination/????-??-??" -exec rm -r -f {} ;
    echo "`date` *** The size of $domain/latest is now `du -sh
```

```
$destination/latest | awk '{print $1}'` ">>$SLOG
    echo -e "`date` *** $domain backup ended">>$SLOG
    echo -e "`date` *** Total allocated `du -sh $destination | awk
'{print $1}'`">>$SLOG
    echo -e "-----"
-----">>$SLOG
done
rm $PID_FILE
```