

Backup filesystem dengan rsync(1)

Wed Feb 25 16:43:34 WIT 2004 oblek

Catatan singkat ini berisikan informasi konfigurasi snapshot backup menggunakan rsync, catatan ini dibuat agar backup dapat dilakukan dengan cepat dan lebih mudah untuk mengantisipasi beberapa hal yang tidak diinginkan (eg. hardware failure, dsb)

Setidaknya dibutuhkan 1 host backup tersendiri dan 1 atau lebih host yang akan dibackup.

Pada kedua host tersebut harus terinstall rsync, pastikan versi tersebut lebih besar dari 2.5.6 (untuk mengantisipasi adanya bugfixes).

Sedikit pengetahuan mengenai ssh, pembuatan passwordless public-key, dan konfigurasinya akan sedikit membantu, silahkan baca manual dari sshd(8) dan ssh_config(5).

Dari filesystem yang akan dibackup dari remote-host, akan diletakkan di backup-host dalam kondisi yang sama persis ketika berada di remote-host, maksudnya, mulai dari owner, timestamp dan hak aksesnya akan persis sama dari asalnya, hal ini digunakan untuk memudahkan transfer file backup dari backup-host ke remote-host tanpa perlu memodifikasi owner, timestamp dan hak aksesnya.

Konfigurasi pertama kali akan dilakukan di backup-host, pertama buatlah passwordless public-key milik user root (bagaimana cara mengantisipasi penyalahgunaan backup menggunakan user ini akan dibahas nanti).

Buat direktori khusus backup untuk per-remote-hosts-nya, eg. /backup/<namahost>, kapasitas direktori ini harus cukup besar untuk menampung file-file backup.

Buat file yang berisikan daftar-daftar file/direktori yang akan dibackup, eg. <namahost>.rsync.list, yang isinya kira-kira spt ini (didalam daftar tsb direktori yang akan dibackup adalah /etc, /root, /usr/local dan /home yang lainnya tidak akan dibackup (ditandai dengan tanda -), untuk lebih lengkap mengenai sintaks file ini silahkan baca manual dari rsync(1)):

```
+ /
+ /etc/
+ /etc/**
+ /usr/
+ /usr/local/
+ /usr/local/**
+ /root/
+ /root/**
+ /home/
+ /home/**
- *
```

Sekarang pada remote-host (yang akan dibackup), buat user dan group rsync (home direktori dan shell dibuat layaknya user biasa).

Modifikasi binari dari rsync di remote-host (di backup host hal ini tidak perlu dilakukan) supaya ownernya (root:rsync), dan permissionnya 4750 (suid bit root), hal ini mutlak perlu dilakukan agar

user rsync (yang login dari backup-host ke remote-host dapat mengeksekusi program rsync di remote-host sebagai user root, mengingat beberapa file-file yang akan dibackup ketika dibaca hanya bisa diakses oleh user root saja (eg. /etc/shadow dsb)).

Kopikan public-key user root di backup-host ke `~rsync/.ssh/authorized_keys`, modifikasi supaya hanya host backup-host saja yang bisa login ke remote-host, dan batasi perintah yang bisa dieksekusi seandainya user rsync login ke remote-host. Berikut ini isi dari file `authorize_keys` setelah dimodifikasi:

```
from="backup-host",command="rsync --server --daemon --
config=/etc/rsyncd.conf ." <public_key_root>
```

Buat file `/etc/rsyncd.conf`, yang berisikan konfigurasi rsync:

```
uid = root
gid = root
use chroot = no
[root]
    path = /
    comment = speak friend and enter
```

Dari backup-host, jalankan program rsync berikut dengan hak akses root (contoh):

```
$ cd /backup/hostA
$ sudo rsync -av -e ssh --exclude-from=hostA.rsync.list rsync@hostA::root .
```

Rotating backup dapat dilakukan untuk mengarsipkan backup sesuai dengan waktu yang ditentukan, untuk lebih jelasnya, silahkan baca artikel yang ada di referensi bawah (Mike Rubel)

```
== potongan script backup ==
#!/bin/sh
# Wed Feb 25 16:49:01 WIT 2004 oblek
#
# Rotating backup script, based on
# http://www.mikerubel.org/computers/rsync_snapshots/
#
# scripts will take 3 arguments,
# hostname, backup instruction
# (hourly, daily, weekly or monthly
# backups) and how many backup archives to keep
#

PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin

_backup_path=/backup

_hostname_fqdn=$1
_instruction=$2
_keep=$3
```

```
_remote_user=rsync
_rsync_module=root

show_usage () {
    echo "Usage: $0 <hostname> <instruction> <keep>"
    echo "where hostname is in fqdn format, eg. foo.bar.org"
    echo "and instruction is: hourly, daily, weekly, monthly"
    echo "while keep is how many archives you want to keep"
    exit 25
}

# FIXME: at the moment only root can do this
check_uid () {
    if [ `id -u` != 0 ]; then
        echo "Only root can do this"
        exit
    fi
}

# check if backup directory exist
check_backup_path () {
    if [ ! -d ${_backup_path} ]; then mkdir -p ${_backup_path}; fi
}

# test valid arguments
check_valid_args () {
    case ${_instruction} in
        hourly) _how_many=`seq ${_keep} 0`;;
        daily) _how_many=`seq ${_keep} 0`;;
        weekly) _how_many=`seq ${_keep} 0`;;
        monthly) _how_many=`seq ${_keep} 0`;;
        *) echo "instruction parameter is not valid"
           show_usage;;
    esac
}

# check if hostname directory exist
check_host_path () {
    # strip fqdn to leading hostname only
    _hostname=`echo ${_hostname_fqdn} | cut -d. -f1`

    if [ ! -d ${_backup_path}/${_hostname}/${_instruction}.0 ]; then
        echo "creating ${_backup_path}/${_hostname}/${_instruction} for this
host"
        echo "don't forget to add new entry in root crontab for this host"
        for i in ${_how_many}
        do mkdir -p ${_backup_path}/${_hostname}/${_instruction}.${i}
        done
    fi
}
}
```

```
# check if hostname.rsync.list is already exist, if not make a default one
check_rsync_exclude_list () {
    if [ ! -f ${_backup_path}/${_hostname}/${_hostname}.rsync.list ]; then
        echo "${_backup_path}/${_hostname}/${_hostname}.rsync.list doesn't
exist, creating default list "
        cat<<EOF > ${_backup_path}/${_hostname}/${_hostname}.rsync.list
+ /
+ /etc/
+ /etc/**
+ /usr/
+ /usr/local/
+ /usr/local/**
+ /root/
+ /root/**
+ /home/
+ /home/**
- *
EOF
    fi
}

#
rotate_snapshot () {
    rm -rf ${_backup_path}/${_hostname}/${_instruction}.`echo ${_how_many} |
cut -d" " -f1`

    for i in ${_how_many}
    do mv ${_backup_path}/${_hostname}/${_instruction}.${(i - 1)} \
        ${_backup_path}/${_hostname}/${_instruction}.${i}
    done
}

#
make_snapshot () {
    rsync \
        -a \
        -e ssh \
        --delete \
        --exclude-from=${_backup_path}/${_hostname}/${_hostname}.rsync.list \
        --link-dest=${_backup_path}/${_hostname}/${_instruction}.1 \
        ${_remote_user}@${_hostname_fqdn}::${_rsync_module}
    ${_backup_path}/${_hostname}/${_instruction}.0
}

#
snapshot () {
    case ${_instruction} in
        hourly)
            rotate_snapshot
            make_snapshot
            ;;
    esac
}
```

```
    daily)
        if [ -d ${_backup_path}/${_hostname}/hourly.0 ]; then
            rotate_snapshot
            cp -al ${_backup_path}/${_hostname}/hourly.0
${_backup_path}/${_hostname}/daily.0
            else
                rotate_snapshot
                make_snapshot
            fi
        ;;
    weekly)
        if [ -d ${_backup_path}/${_hostname}/daily.0 ]; then
            rotate_snapshot
            cp -al ${_backup_path}/${_hostname}/daily.0
${_backup_path}/${_hostname}/weekly.0
            else
                rotate_snapshot
                make_snapshot
            fi
        ;;
    monthly)
        if [ -d ${_backup_path}/${_hostname}/weekly.0 ]; then
            rotate_snapshot
            cp -al ${_backup_path}/${_hostname}/weekly.0
${_backup_path}/${_hostname}/monthly.0
            else
                rotate_snapshot
                make_snapshot
            fi
        ;;
    esac
}

# main
check_uid
check_backup_path
check_valid_args
check_host_path
check_rsync_exclude_list
snapshot
```

Referensi:

- [Rsync, Rsync Homepage](#)
- [DevShed, File Synchronization With Rsync](#)
- [Mike Rubel, Easy Automated Snapshot-Style Backups with Linux and Rsync](#)

From:

<http://wiki.corebsd.or.id/> - **CoreBSD Wiki**

Permanent link:

<http://wiki.corebsd.or.id/doku.php/coreartikel:backupdengansync>

Last update: **2025/10/25 17:09**

