# Sauvegarde

Démonstration pour réaliser des sauvegardes avec une récurrence

- Backup avec rsnapshot
- Exemple configuration rsnapshot

# Backup avec rsnapshot

## Présentation de `rsnapshot`

**rsnapshot** permet de conserver des fichiers, dans des versions différentes capturées dans le temps.

Pour du texte ou du code informatique, c'est aussi ce que font des gestionnaires de version, comme Git ou SVN. Et ils le font mieux : ces gestionnaires de version capturent chaque version d'un fichier lorsqu'il est modifié, et proposent des fonctionnalités avancées de fusion, de récupération d'une portion spécifique du texte, etc.

En comparaison **rsnapshot** a un intérêt particulier quand on travaille sur des fichiers multimédia par exemple (Git n'est pas efficace du tout pour sauvegarder plusieurs versions d'une vidéo, il ferait perdre un énorme espace de stockage).
Par conséquent **rsnapshot** est très utile pour conserver des versions de fichiers indépendamment de leur type. C'est donc l'idéal pour sauvegarder son espace `home` par exemple.

Par rapport aux outils ordinaires de sauvegarde, disposer d'instantanés permet de retrouver une version antérieure, sauvegardée à une date spécifique, d'un document. Une sauvegarde ordinaire ne protège pas d'une bêtise enregistrée, mais seulement d'une perte accidentelle.

# Comment mettre en place une sauvegarde avec `rsnapshot` ?

## Prérequis

Installation de `rsnapshot`

```
sudo apt-get install rsnapshot
```

## Configuration

Réaliser une sauvegarde du fichier de configuration de `rsnapshot` avant de le modifier pour éviter toutes fausses manip sur le fichier :

```
cp /etc/rsnapshot.conf /etc/rsnapshot.conf.backup
```

Maintenant on peut éditer le fichier `/etc/rsnapshot.conf` :

```
nano /etc/rsnapshot.conf
```

(voir l'exemple de configuration rsnapshot.conf : Exemple de configuration)

Vérifier la syntaxe de la configuration :

```
rsnapshot configtest
```

## Test

Pour tester votre configuration utiliser l'option -t suivi de l'intervalle choisi :

```
rsnapshot -t daily
```

Cette option va vous montrer la liste des commandes qui vont être exécutées en fonction de votre configuration du fichier.

## Exécuter

Pour exécuter manuellement la création d'un instantané, qui déclenche aussi leur rotation pour un intervalle donné :

```
rsnapshot daily
```

> Avant de pouvoir sauvegarder un niveau supérieur, il faut que tout le niveau inférieur soit complet.
> Exemple :

```
sudo rsnapshot weekly
/var/cache/rsnapshot/weekly.6 not present (yet), nothing to copy
sudo rsnapshot monthly
/var/cache/rsnapshot/monthly.23 not present (yet), nothing to copy
```

# Automatiser

Il est fortement conseiller d'automatiser l'exécution de `rsnapshot` afin d'être sur que les données sauvegarder soit à jours des derniers changements.

Dans notre cas nous allons utiliser le `crontable` pour automatiser son exécution régulièrement. Il existe d'autres façons de faire mais nous allons présenter seulement notre cas.

## Configuration du `crontable`

Pour automatiser, il faut alors décommenter les lignes du fichier **`/etc/cron.d/rsnapshot`**

```
# This is a sample cron file for rsnapshot.
# The values used correspond to the examples in /etc/rsnapshot.conf.
# There you can also set the backup points and many other things.
#
# To activate this cron file you have to uncomment the lines below.
# Feel free to adapt it to your needs.

# 0 */4         * * *           root    /usr/bin/rsnapshot alpha
30 3    * * *           root    /usr/bin/rsnapshot daily
# 0  3          * * 1           root    /usr/bin/rsnapshot weekly
# 30 2          1 * *           root    /usr/bin/rsnapshot monthly
```

En résumé:

- alpha se lance toutes les quatre heures tout le temps au heures exactes à partir de 0 heure.
- daily se lance tous les jours à 3 heures trente.
- weekly se lance tous les lundi à 0 heure 30.
- monthly se lance tous les premiers du mois à 2 heures 30.

Nous avons seulement dé-commenté la ligne `daily` car nous souhaitons mettre à jours notre sauvegarde tous les jours. Vous pouvez dé-commenter la ligne qui correspond à votre besoin (toutes les heures, tous les jours, toutes les semaines, tous les mois).

# Consulter

Il peut être intéressant de consulter la sauvegarde pour comparer la version des fichiers ou tout simplement pour restaurer la sauvegarde.

La sauvegarde la plus récente est disponible **sous forme de duplication** dans la structure **`/{snapshot_root}/daily.0/{backup_folder}`**.

Dans notre cas, en fonction de notre configuration du fichier, la structure est la suivante : `/mydisk/mediacloud/backup/daily.0/nextcloud`.

Ce qui donne :

```
$ ls -l /mydisk/mediacloud/backup/daily.0/nextcloud/
total 21764
drwxr-xr-x  7 user user     4096  3 oct.   2022 john
drwxr-xr-x  7 user user     4096 29 déc.  18:51 mike
drwxr-xr-x  7 user user     4096 12 mai    2023 user
drwxr-xr-x  7 user user     4096  2 juin   2023 franck
```

Les sauvegardes précédentes sont incrémentées de 1. La plus ancienne a le N° le plus élevé
L'épuration est automatiquement faite en fonction de la périodicité :

<afficher la liste des daily.* et weekly.*>

> A savoir : **Chaque sauvegarde est complète pour chaque dossier. Mais les fichiers ne sont pas dupliqués entre chaque incrémentation.** Il est utilisé des liens durs et non symboliques.

# Exemple configuration rsnapshot

> ATTENTION: Le fichier ne doit pas contenir d'espace !!!
> Seul les tabulations sont autorisées.

```
#################################################
# rsnapshot.conf - rsnapshot configuration file #
#################################################
#                                               #
# PLEASE BE AWARE OF THE FOLLOWING RULE:        #
#                                               #
# This file requires tabs between elements      #
#                                               #
#################################################

#######################
# CONFIG FILE VERSION #
#######################

config_version	1.2

###########################
# SNAPSHOT ROOT DIRECTORY #
###########################

# All snapshots will be stored under this root directory.
#
snapshot_root	/mydisk/mediacloud/backup/

# If no_create_root is enabled, rsnapshot will not automatically create the
# snapshot_root directory. This is particularly useful if you are backing
# up to removable media, such as a FireWire or USB drive.
#
#no_create_root	1
```

```
#################################
# EXTERNAL PROGRAM DEPENDENCIES #
#################################

# LINUX USERS:   Be sure to uncomment "cmd_cp". This gives you extra features.
# EVERYONE ELSE: Leave "cmd_cp" commented out for compatibility.
#
# See the README file or the man page for more details.
#
cmd_cp	/bin/cp

# uncomment this to use the rm program instead of the built-in perl routine.
#
cmd_rm	/bin/rm

# rsync must be enabled for anything to work. This is the only command that
# must be enabled.
#
cmd_rsync	/usr/bin/rsync

# Uncomment this to enable remote ssh backups over rsync.
#
cmd_ssh	/usr/bin/ssh

# Comment this out to disable syslog support.
#
cmd_logger	/usr/bin/logger

# Uncomment this to specify the path to "du" for disk usage checks.
# If you have an older version of "du", you may also want to check the
# "du_args" parameter below.
#
cmd_du	/usr/bin/du

# Uncomment this to specify the path to rsnapshot-diff.
#
#cmd_rsnapshot_diff	/usr/bin/rsnapshot-diff

# Specify the path to a script (and any optional arguments) to run right
# before rsnapshot syncs files
```

```
#
#cmd_preexec	/path/to/preexec/script

# Specify the path to a script (and any optional arguments) to run right
# after rsnapshot syncs files
#
#cmd_postexec	/path/to/postexec/script

# Paths to lvcreate, lvremove, mount and umount commands, for use with
# Linux LVMs.
#
#linux_lvm_cmd_lvcreate	/sbin/lvcreate
#linux_lvm_cmd_lvremove	/sbin/lvremove
#linux_lvm_cmd_mount	/bin/mount
#linux_lvm_cmd_umount	/bin/umount

#########################################
#     BACKUP LEVELS / INTERVALS         #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc.         #
#########################################

#sauvegarde tous les jours
retain	daily	7

#Sauvegarde toutes les semaines
retain	weekly	4

#Sauvegarde tous les mois
retain	monthly	12

##############################################
#              GLOBAL OPTIONS              #
# All are optional, with sensible defaults #
##############################################

# Verbose level, 1 through 5.
# 1     Quiet           Print fatal errors only
# 2     Default         Print errors and warnings only
# 3     Verbose         Show equivalent shell commands being executed
# 4     Extra Verbose   Show extra verbose information
```

```
# 5     Debug mode      Everything
#
verbose  3

# Same as "verbose" above, but controls the amount of data sent to the
# logfile, if one is being used. The default is 3.
# If you want the rsync output, you have to set it to 4
#
loglevel 3

# If you enable this, data will be written to the file you specify. The
# amount of data written is controlled by the "loglevel" parameter.
#
#logfile /var/log/rsnapshot.log

# If enabled, rsnapshot will write a lockfile to prevent two instances
# from running simultaneously (and messing up the snapshot_root).
# If you enable this, make sure the lockfile directory is not world
# writable. Otherwise anyone can prevent the program from running.
# chown permission to /mnt/run/ to user
lockfile /mnt/run/rsnapshot.pid

# By default, rsnapshot check lockfile, check if PID is running
# and if not, consider lockfile as stale, then start
# Enabling this stop rsnapshot if PID in lockfile is not running
#
#stop_on_stale_lockfile  0

# Default rsync args. All rsync commands have at least these options set.
#
#rsync_short_args -a
rsync_short_args -ravh
#rsync_long_args --delete --numeric-ids --relative --delete-excluded
rsync_long_args  --delete --numeric-ids

# ssh has no args passed by default, but you can specify some here.
#
#ssh_args -p 22

# Default arguments for the "du" program (for disk space reporting).
# The GNU version of "du" is preferred. See the man page for more details.
```

```
# If your version of "du" doesn't support the -h flag, try -k flag instead.
#
#du_args□-csh

# If this is enabled, rsync won't span filesystem partitions within a
# backup point. This essentially passes the -x option to rsync.
# The default is 0 (off).
#
#one_fs□□0

# The include and exclude parameters, if enabled, simply get passed directly
# to rsync. If you have multiple include/exclude patterns, put each one on a
# separate line. Please look up the --include and --exclude options in the
# rsync man page for more details on how to specify file name patterns.
#
#include□???
#include□???
#exclude□???
#exclude□???

# The include_file and exclude_file parameters, if enabled, simply get
# passed directly to rsync. Please look up the --include-from and
# --exclude-from options in the rsync man page for more details.
#
#include_file□/path/to/include/file
#exclude_file□/path/to/exclude/file

# If your version of rsync supports --link-dest, consider enabling this.
# This is the best way to support special files (FIFOs, etc) cross-platform.
# The default is 0 (off).
#
#link_dest□0

# When sync_first is enabled, it changes the default behaviour of rsnapshot.
# Normally, when rsnapshot is called with its lowest interval
# (i.e.: "rsnapshot alpha"), it will sync files AND rotate the lowest
# intervals. With sync_first enabled, "rsnapshot sync" handles the file sync,
# and all interval calls simply rotate files. See the man page for more
# details. The default is 0 (off).
#
#sync_first□0
```

```
# If enabled, rsnapshot will move the oldest directory for each interval
# to [interval_name].delete, then it will remove the lockfile and delete
# that directory just before it exits. The default is 0 (off).
#
#use_lazy_deletes	0

# Number of rsync re-tries. If you experience any network problems or
# network card issues that tend to cause ssh to fail with errors like
# "Corrupted MAC on input", for example, set this to a non-zero value
# to have the rsync operation re-tried.
#
#rsync_numtries 0

# LVM parameters. Used to backup with creating lvm snapshot before backup
# and removing it after. This should ensure consistency of data in some special
# cases
#
# LVM snapshot(s) size (lvcreate --size option).
#
#linux_lvm_snapshotsize	100M

# Name to be used when creating the LVM logical volume snapshot(s).
#
#linux_lvm_snapshotname	rsnapshot

# Path to the LVM Volume Groups.
#
#linux_lvm_vgpath	/dev

# Mount point to use to temporarily mount the snapshot(s).
#
#linux_lvm_mountpath	/path/to/mount/lvm/snapshot/during/backup

################################
### BACKUP POINTS / SCRIPTS ###
################################

# LOCALHOST
#backup	/home/		localhost/
#backup	/etc/		localhost/
```

```
#backup□/usr/local/□localhost/
#backup□/var/log/rsnapshot□□localhost/
#backup□/etc/passwd□localhost/
#backup□/home/foo/My Documents/□□localhost/
#backup□/foo/bar/□localhost/□one_fs=1,rsync_short_args=-urltvpog
#backup_script□/usr/local/bin/backup_pgsql.sh□localhost/postgres/
# You must set linux_lvm_* parameters below before using lvm snapshots
#backup□lvm://vg0/xen-home/□lvm-vg0/xen-home/

# EXAMPLE.COM
#backup_exec□/bin/date "+ backup of example.com started at %c"
#backup□root@example.com:/home/□example.com/□+rsync_long_args=--bwlimit=16,exclude=core
#backup□root@example.com:/etc/□example.com/□exclude=mtab,exclude=core
#backup_exec□ssh root@example.com "mysqldump -A > /var/db/dump/mysql.sql"
#backup□root@example.com:/var/db/dump/□example.com/
#backup_exec□/bin/date "+ backup of example.com ended at %c"

# CVS.SOURCEFORGE.NET
#backup_script□/usr/local/bin/backup_rsnapshot_cvsroot.sh□rsnapshot.cvs.sourceforge.net/

# RSYNC.SAMBA.ORG
#backup□rsync://rsync.samba.org/rsyncftp/□rsync.samba.org/rsyncftp/

# on choisi qu'est ce qu'on doit backup et vers ou
# dans notre cas on veut backup /mydisk/cloud/datacloud/nextcloud/data/ provenant du serveur
configssh
# vers le dossier nextcloud de notre racine (snapshot_root)
backup□configssh:/mydisk/cloud/datacloud/nextcloud/data/□nextcloud/
```