Backup and Restore

Valeria Mazzola
mazzolavale1@gmail.com

Corsi GNU/Linux Avanzati 2016

What are backups?

Definition

The copying and archiving of computer data so that it may be used to restore the original after a data loss event.

Why backups?

Bad things can and do happen

- A disk can break if dropped
- or shaken...
- ...or will just die of old age

What (not) to back up

Do:

- /home

Maybe:

- /etc
- /var

Don't:

- /proc /sys /tmp
- /dev /mnt /media

Backup types

Backups can be

- **full** - a backup of all the files and folders
- **incremental** - an incremental backup contains all the differences since the last *incremental* backup
- **differential** - a differential backup contains the changes since the last *full* backup

Where do I put my data?

- Optical supports: CDs, DVDs, Blu-ray
- Hard disks (HDD)
- Solid-state drives (SSD)
- Flash drives
- Cloud storage services

dd

dd is a simple tool that can be used for disk cloning

- `dd if=/dev/sdX of=/dev/sdY && sync`
    - **if**: input file
    - **of**: output file

GNU ddrescue

gddrescue copies data from one file or block device to another, trying hard to rescue the good parts first in case of read errors

- `ddrescue [options] /dev/sdX outfile mapfile`
  - **mapfile:** a human readable text file ddrescue uses to manage the copy

warning

for the rescued data to be correct, both dd and gddrescue are best used on unmounted devices

GNU ddrescue

gddrescue copies data from one file or block device to another, trying hard to rescue the good parts first in case of read errors

- `ddrescue [options] /dev/sdX outfile mapfile`
  - **mapfile:** a human readable text file ddrescue uses to manage the copy

warning

for the rescued data to be correct, both dd and gddrescue are best used on unmounted devices

rsync

also known as an advanced version of cp

Pros

- (unlike cp) preserves hard and symbolic links, file permissions and ownership, modification times, etc
- designed to be network efficient: only transfers file changes
- easy to use

Cons

- no storage encryption

rsync: usage

- `$ rsync -Pr source destination`
  - **P**: keep partially transferred files if the transfer is interrupted
  - **r**: recursive directory option

- `$ rsync source host:destination`
  - uses ssh by default, but can also be forced with the `-e ssh` option

- `$ rsync -aAXv --exclude={...} /* /backup_folder`
  - backup /* while following symlinks and preserving file properties

rsync: usage

- `$ rsync -Pr source destination`
  - **P**: keep partially transferred files if the transfer is interrupted
  - **r**: recursive directory option

- `$ rsync source host:destination`
  - uses ssh by default, but can also be forced with the **`-e ssh`** option

- `$ rsync -aAXv --exclude={...} /* /backup_folder`
  - backup /* while following symlinks and preserving file properties

rsync: usage

- `$ rsync -Pr source destination`
  - **P**: keep partially transferred files if the transfer is interrupted
  - **r**: recursive directory option

- `$ rsync source host:destination`
  - uses ssh by default, but can also be forced with the **`-e ssh`** option

- `$ rsync -aAXv --exclude={...} /* /backup_folder`
  - backup /* while following symlinks and preserving file properties

rsnapshot: rsync automated

rsnapshot produces automated, periodical system snapshots

Pros

- preserves hard and symbolic links, file permissions and ownership, modification times, etc
- network efficient
- each snapshot contains a full system backup
- easy to use

Cons

- no storage encryption

duplicity

duplicity produces encrypted, incremental backups in tar format

Pros

- preserves hard and symbolic links, file permissions and ownership, modification times, etc
- network efficient
- incremental backups
- supports storage encryption with gpg
- easy to use

duplicity: usage

- `duplicity /home/me scp://user@host//some/directory`

- `duplicity [restore] scp://user@host//some/directory /home/me`

- `duplicity full /home/me scp://user@host//some/directory`

duplicity: usage

- `duplicity /home/me scp://user@host//some/directory`
- `duplicity [restore] scp://user@host//some/directory /home/me`
- `duplicity full /home/me scp://user@host//some/directory`

duplicity: usage

- `duplicity /home/me scp://user@host//some/directory`
- `duplicity [restore] scp://user@host//some/directory /home/me`
- `duplicity full /home/me scp://user@host//some/directory`

duplicity: usage

- `duplicity list-current-files scp://user@host//some/directory`
  - list the files contained in the backup
- `duplicity [restore] -t 3D scp://user@host//some/directory /home/me`
  - **-t:** specify the time from which to restore files
- `duplicity remove-older-than 30D scp://user@host//some/directory`

duplicity: usage

- `duplicity list-current-files scp://user@host//some/directory`
  - list the files contained in the backup

- `duplicity [restore] -t 3D scp://user@host//some/directory /home/me`
  - **-t:** specify the time from which to restore files

- `duplicity remove-older-than 30D scp://user@host//some/directory`

duplicity: usage

- `duplicity list-current-files scp://user@host//some/directory`
  - list the files contained in the backup
- `duplicity [restore] -t 3D scp://user@host//some/directory /home/me`
  - **-t:** specify the time from which to restore files
- `duplicity remove-older-than 30D scp://user@host//some/directory`

rdiff-backup

another utility for incremental backups

Pros

- preserves hard and symbolic links, file permissions and ownership, modification times, etc
- network efficient
- incremental backups
- easy to use

Cons

- no storage encryption

Demo

Demo!

One last thing

- Remember to keep a copy of your private key!
  - without it, you won't be able to decrypt your own backups :)

References

- https://en.wikipedia.org/wiki/Incremental_backup
- https://en.wikipedia.org/wiki/Differential_backup
- https://wiki.archlinux.org/.../Full_system_backup_with_rsync
- https://wiki.archlinux.org/index.php/Duplicity
- http://duplicity.nongnu.org
- http://www.nongnu.org/rdiff-backup/index.html
- https://www.digitalocean.com/community/tutorials/how-to-use-duplicity-with-gpg-to-securely-automate-backups-on-ubuntu
- https://blog.roundside.com/duplicity-vs-rdiff-backup-in-action/
- Unix and Linux system administration handbook, fourth edition, Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley

Thank you

Thank you!

https://www.poul.org