Organizing backups

Jérôme Dockès

Files we don’t want to lose

- Code, PhD thesis, data, notes, admnistrative stuff
- Pictures
- Grandma’s chocolate cake recipe
- Music recordings
- . . .

They can easily be lost

- Overwrite or delete a file by mistake ⋆
- Mess up the filesystem (e.g. during an OS upgrade)
- Hardware failure ⋆
- Stolen laptop or phone ⋆
- Malware
- Something involving coffee
- Something involving a child or a pet
- . . .

⋆ happened to me

Small experiments

Stolen bag

- Turn off phone
- Borrow someone else’s computer or boot from a USB key (don’t mount our drive)
- Do we still have:
  - Work we were doing before lunch
  - Photos of our last holidays
  - Grandma’s chocolate cake recipe
- Are they easy to recover?

Deleted file

- Delete a file
- Come back a week later; can we recover it?

These experiments will catch problems like

- I don’t have backups
- I have backups <somewhere> but my passphrase or ssh key to access <somewhere> was (only) on the stolen laptop
- I still have the main branch of my project but not the new feature branch I hadn’t pushed to the remote repo yet
- I only have the photos I shared with my friends but some good ones are missing

Remember: we all need backups!

- Failures **will** happen
- **I** am responsible for backing up my files – and shouldn’t count on someone else to do it for me
- I must **check** regularly that I can recover from backups
- By the time I need backups it will be **too late** to do anything if I don’t have them

Things that kind of look like backups but aren’t

Tools for synchronization if they only keep the current version:
e.g. plain `rsync`

- "If machine A breaks I have the files on machine B" . . . :) ?
- Likely scenario:
  - Accidentally overwrite file on machine A
  - Sync; change gets reflected on machine B
  - The end.

We need a **history** of snapshots

Things that kind of look like backups but aren’t

Remote git repos (e.g. on GitHub)

- Not every file is included by default
- Not automated nor scheduled
- Not made for backups but for sharing, publishing
- Does not work for binary files (no deduplication)
- Not encrypted by default
- Not very easy to prune old versions
- Most free hosting services will have small limits on repo and file sizes

We need a **systematic** and **dedicated** solution

So what do we want from our backup strategy?

- Convenient (at least once it is set up), one solution for all types of files
- Automated or at least periodically scheduled
- All files backed up by default; can choose which to exclude
- Private, i.e. encrypted by default
  - I won’t share Grandma’s recipe with GitHub just to back it up
- Keep history of previous versions (spaced in a sensible way)
- Space efficient
  - Good deduplication
  - Possibly compression
- Easy to check
  - "Restore happens rarely, so should be possible but not necessarily easy" – **False**
  - Restoring should be easy because I need to check regularly that it works as intended

Good news

- We can get all that very easily
  - and for free or cheap if data is not huge
- Today we will see one approach but there are probably a lot of better ones
- See interactive session:
  - Restic or Borg
  - Cron
  - ssh agent, keychain, gpg agent

Conclusion

- We should make sure what we need to access the backups is backed up separately
  - e.g. a backup of the repository passphrase that is locked inside the repository does not help
- Check from time to time that everything is working properly
- Curious to hear about other approaches to backups used in the lab!