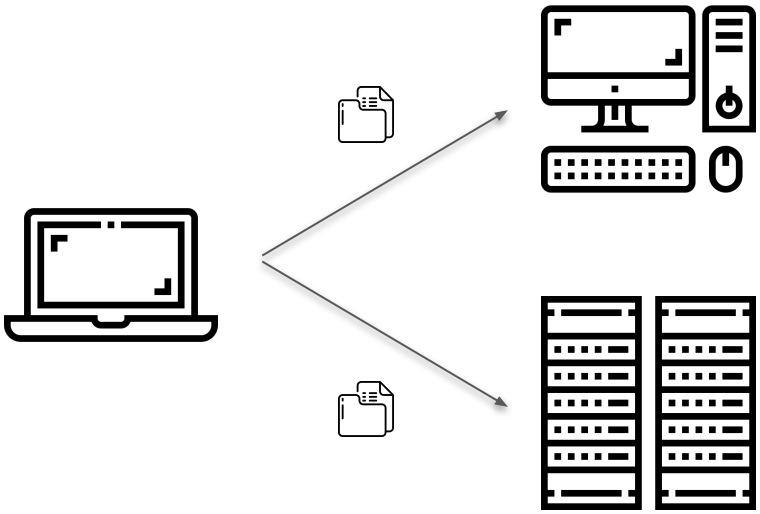
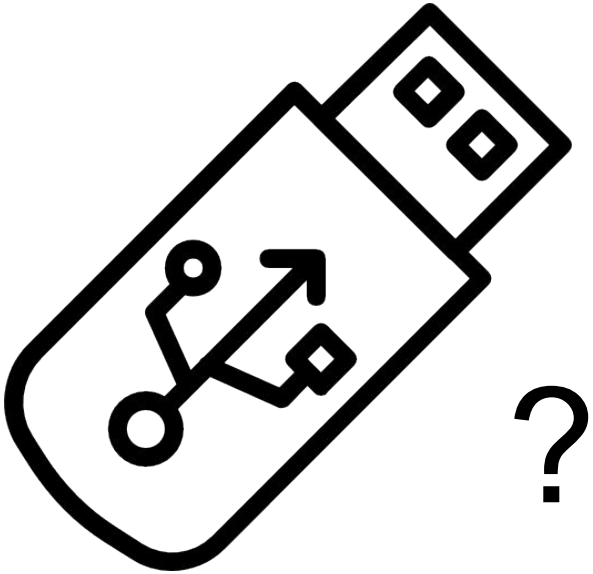




- ✕ Need physical access to the machine
- ✕ Inconvenient / slow
- ✕ Hardware limitations (space, etc.)

```
scp username@hostname:/path/to/remote/file /path/to/local/file
```

- ✓ Don’t need physical access to the machine
- Custom synchronization rules (manual)
- ✗ Terrible UI

- ✗ Data privacy
- ✗ ~~Hardware~~ Service limitations (space, etc.)
- ≈ Custom synchronization rules (manual)
- ✓ Nice UI
- ✓ Sharing files with multiple users

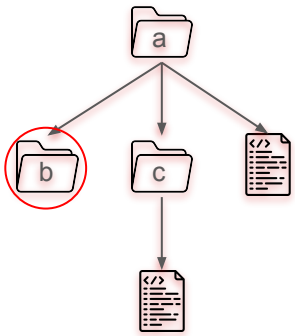
?

```
sudo sshfs username@hostname:/remote/directory /local/directory
```

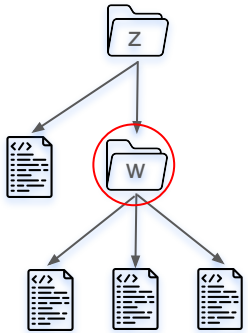
(most file managers support SSHFS!)

- Custom synchronization rules (manual)
- ✓ Interactive, can use your file manager!

john-laptop

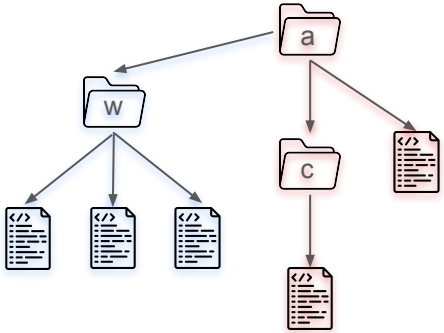


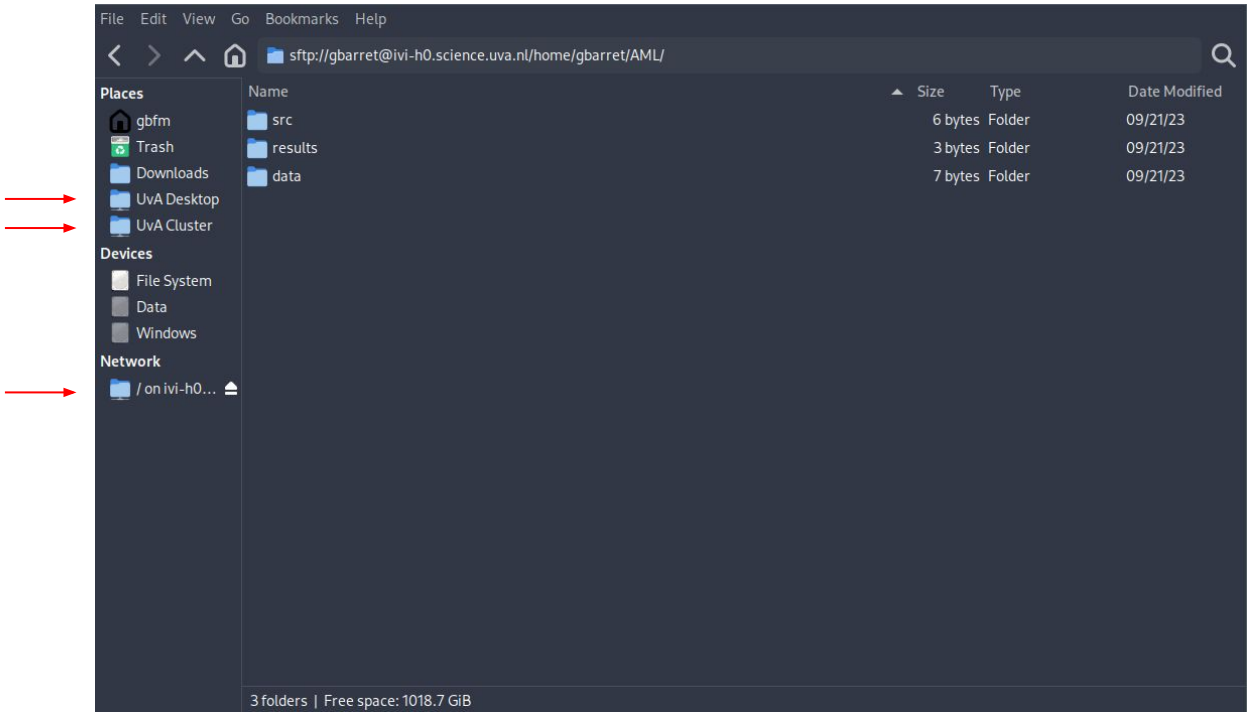
john-desktop



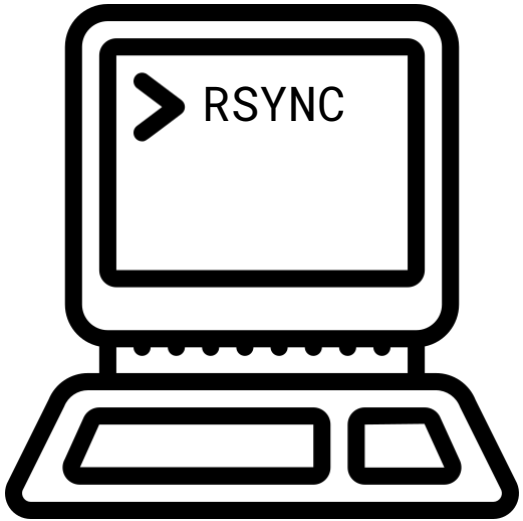

```
sudo sshfs john@john-desktop:/z/w /a/b
```

john_laptop




File Edit View Go Bookmarks Help
sftp://gbarret@ivi-h0.science.uva.nl/home/gbarret/AML/
Places
gbfm
Trash
Downloads
UvA Desktop
UvA Cluster
Devices
File System
Data
Windows
Network
/ on ivi-h0...
Name
Size
Type
Date Modified
src
6 bytes
Folder
09/21/23
results
3 bytes
Folder
09/21/23
data
7 bytes
Folder
09/21/23
3 folders | Free space: 1018.7 GiB

```
rsync -z -a --delete --progress -e ssh username@hostname:/remote/directory /local/directory
```



- ✓ Custom synchronization rules!
- ✗ Not interactive

```
rsync -z -a --delete --progress -e ssh username@hostname:/remote/directory /local/directory
                                                    (place B)                 (place A)
```

Copy from place A to place B

- `-z` : compress before transfer, decompress after transfer
- `-a` : archive mode (recursive copy, preserve modification times, permissions, owner)
- `--delete` : delete while files no longer in place A
- `--progress` : show a progress bar

Versioning!?

- ✓ Data privacy
- ✓ Custom synchronization rules
- ✓ Sharing files with multiple users
- ✓ Versioning!
- ✓ Everyone uses it
- ✗ Doesn’t handle large files well
- ✗ Too much overhead for small transfers

`git clone https://github.com/owner/repo.git`	download repository files from remote server
`cd repo`	go into repository directory
`git log`	see repository history
`git branch feature2`	create new branch named feature2
`git checkout feature2`	move to branch feature2
`(... do some work)`	(...)
`git add file1.txt file2.txt`	add new or modified files to staging area
`(... do more work)`	(...)
`git status`	shows current repo and file status
`git commit -m "Corrected ponctuation"`	create a commit with a custom message
`git checkout master`	move to master branch
`git pull`	get changes from a remote repository
`git merge feature2`	merge changes from feature2 into master branch
`git push`	send changes to remote repository

Some git best practices & tips:

- No large files
- Frequent commits
- Atomic commits
- Good commit messages
- Keep the tree clean
- Learn the command line tool!

Thank you