

Git Basic Concept



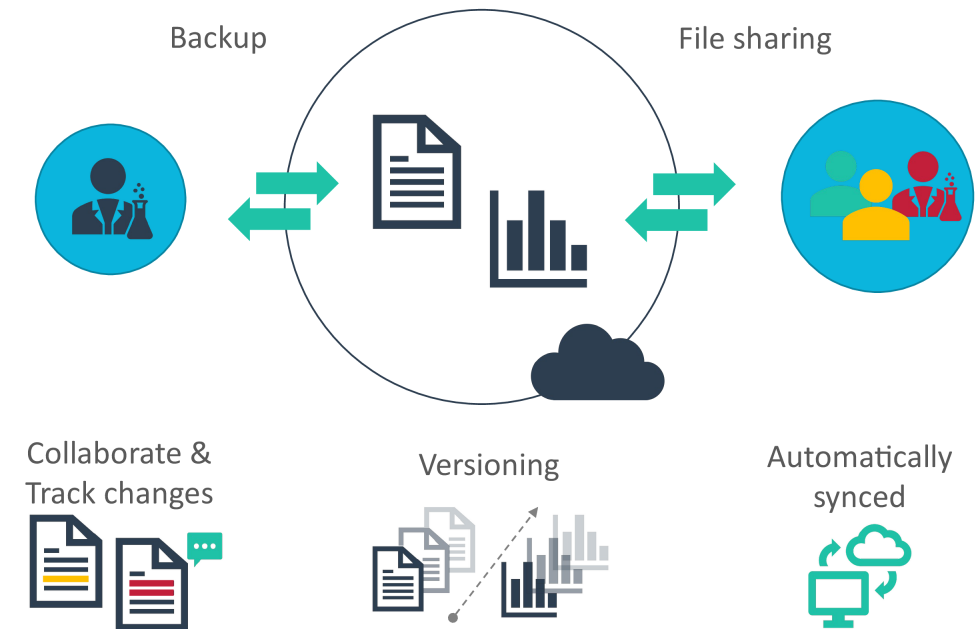
Cloud Services

- ✓ Documents
- ✓ Small data
- ✓ Presentations

X Code

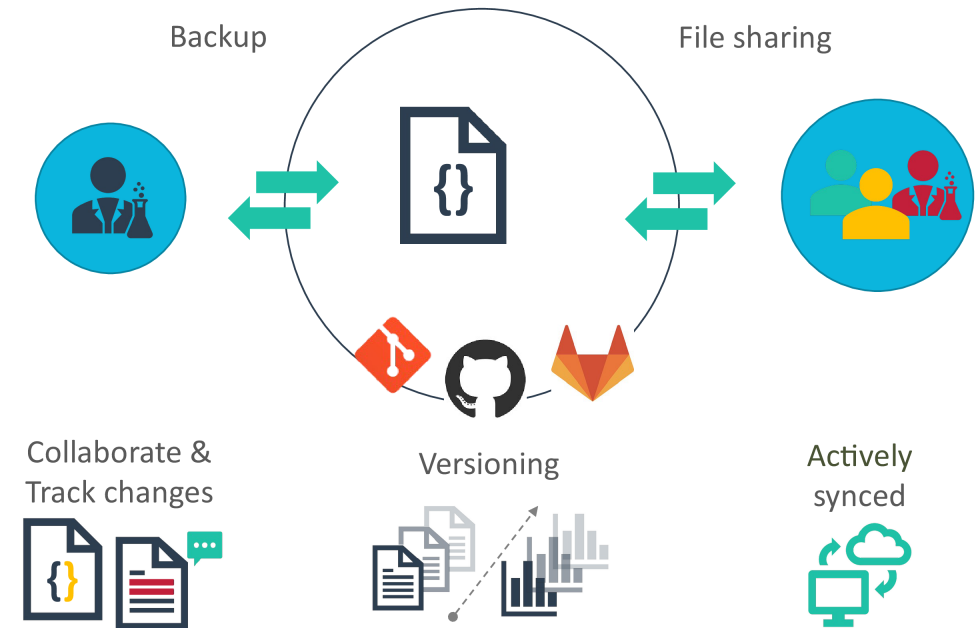
X Data analytical projects

X Big ("raw") data



Git and git platforms

- ~ Documents
- ✓ Small data
- ~ Presentations
- ✓ ✓ Code
- ✓ ✓ Data analytical projects
- ~ Big ("raw") data



Why git? \approx > Why code?

- Save time
- Avoid doing repetitive tasks “by hand”
- Reuse scripts, analyses, pipelines
- Reproduce results



Git: summary

- Version control system
- Git "repository" = a central data package (directory)
- Allows to track changes to any file in the repository
 - **What** was changed
 - **When** was it changed
 - By **whom** was it changed
 - **Why** was it changed?



GitHub and GitLab

- A well-documented cloud environment
- Active syncing
- Not automatically synced
- Non-automated version control
- You have the control what changes to track and what to sync
- Time machine to go back to older versions



GitHub and Gitlab team projects

Simplifies concurrent work & merging changes

- Online service to host our projects
- Share code with other developers
- Others can download our projects, work on and contribute to them
- They can upload their changes and merge them with the main project



Cloud vs. Git

Track changes



Collaboration



Versioning



Syncing



Access



Data security



Cloud services



- ✓ Documents
- ✓ Small data
- ✓ Presentations

Automated

Automated

Oftentimes only within
organization / institution

Private / commercial

Git / GitHub / GitLab



- ✓ Code
- ✓ Data analytical projects

issue tracker, tracked contribution

Well-documented
(commit history)

Active / controlled
by user

Easily collaborate
across institutions

GitLab: on-premise
and custom
solutions

