



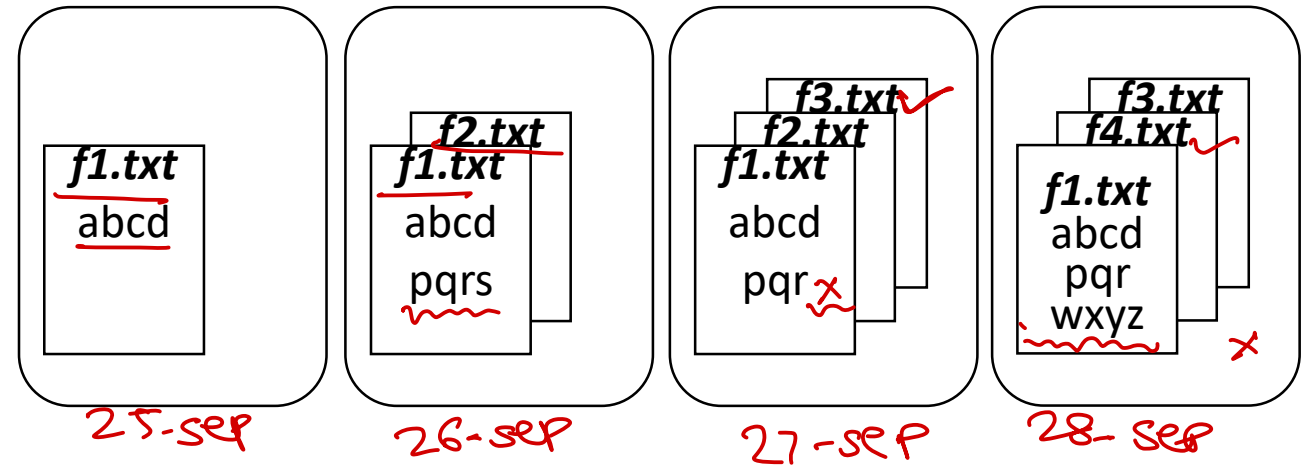
# GIT

*Trainer: Nilesh Ghule.*



# Version control system

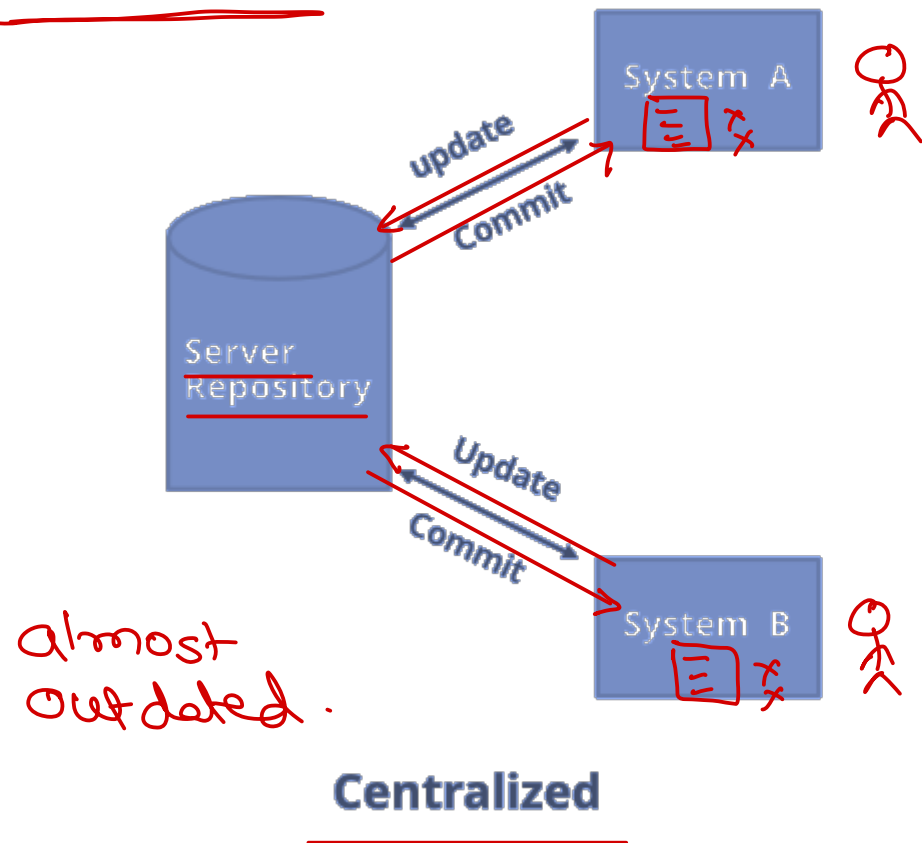
- For management of documents/source code
- Logical way to organize and control revisions of the code.
- Tracks/controls changes in code.
- Also known as
  - ✓ Revision control system
  - ✓ Source code control system
- Two types:
  - Centralized VCS
    - e.g. CVS, SVN, Bazaar, ...
  - Distributed VCS
    - e.g. Git, Mercurial, Arch, ...



# Version control system

## Centralized VCS

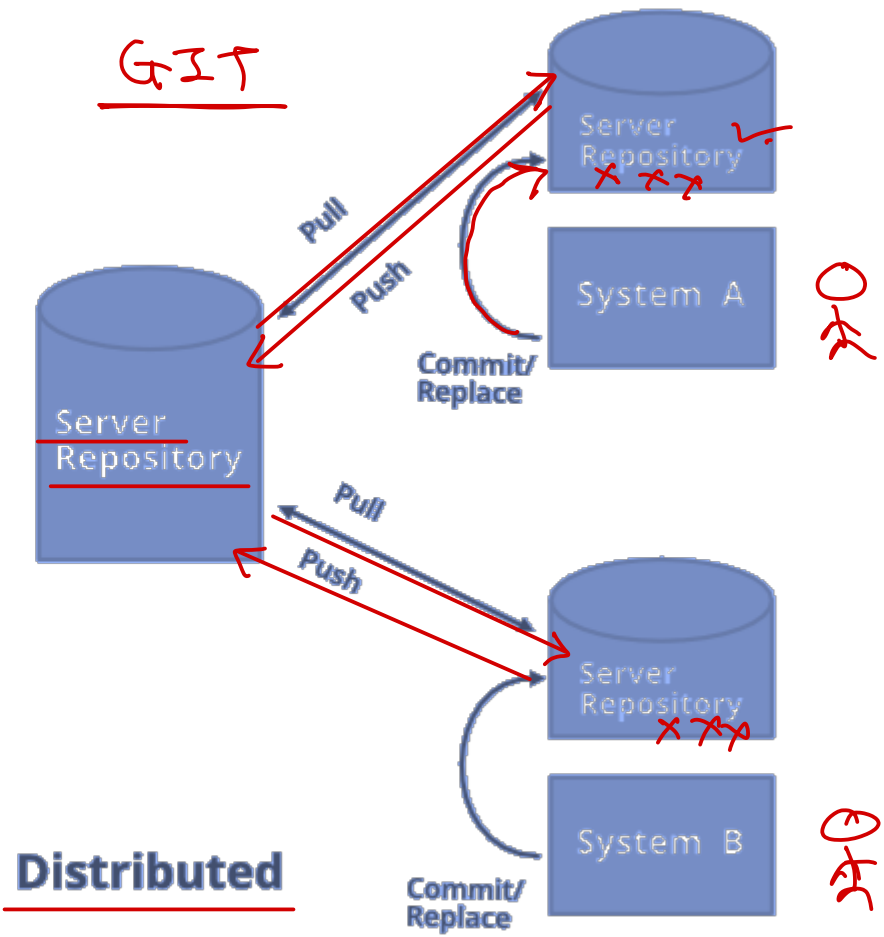
winCVS



almost out dated.

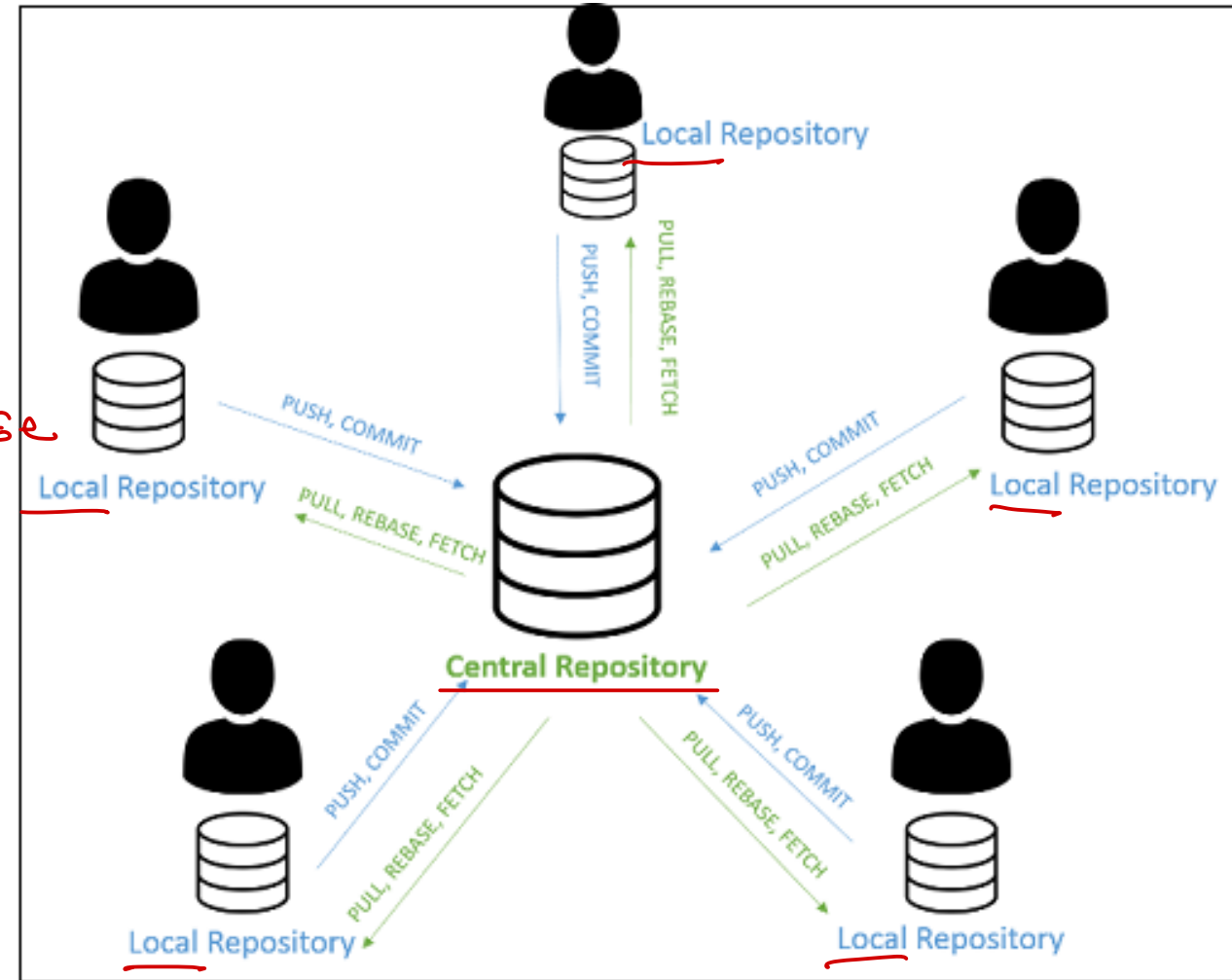
## Distributed VCS

GIT



# GIT

- Distributed VCS & SCM.
- Designed by developed by Linus Torvalds to manage Linux kernel source code.
- Open source software.
- Free under GPL. *General Public License*
- Development
  - Began on 3-Apr-2005.
  - Announced on 6-Apr.
  - Became self-hosing on 7-Apr.
  - First branch merging on 18-Apr.
  - Achieved performance goals on 29-Apr.
  - Began kernel (2.6.12) management from 16-Jun.



# Terminologies

- GIT repository

- Directory containing code and its metadata (.git).

- GIT working

- Working area
- Staging area
- Repository

- Commits

- Hold a state of repository.
- GIT maintains commit history.

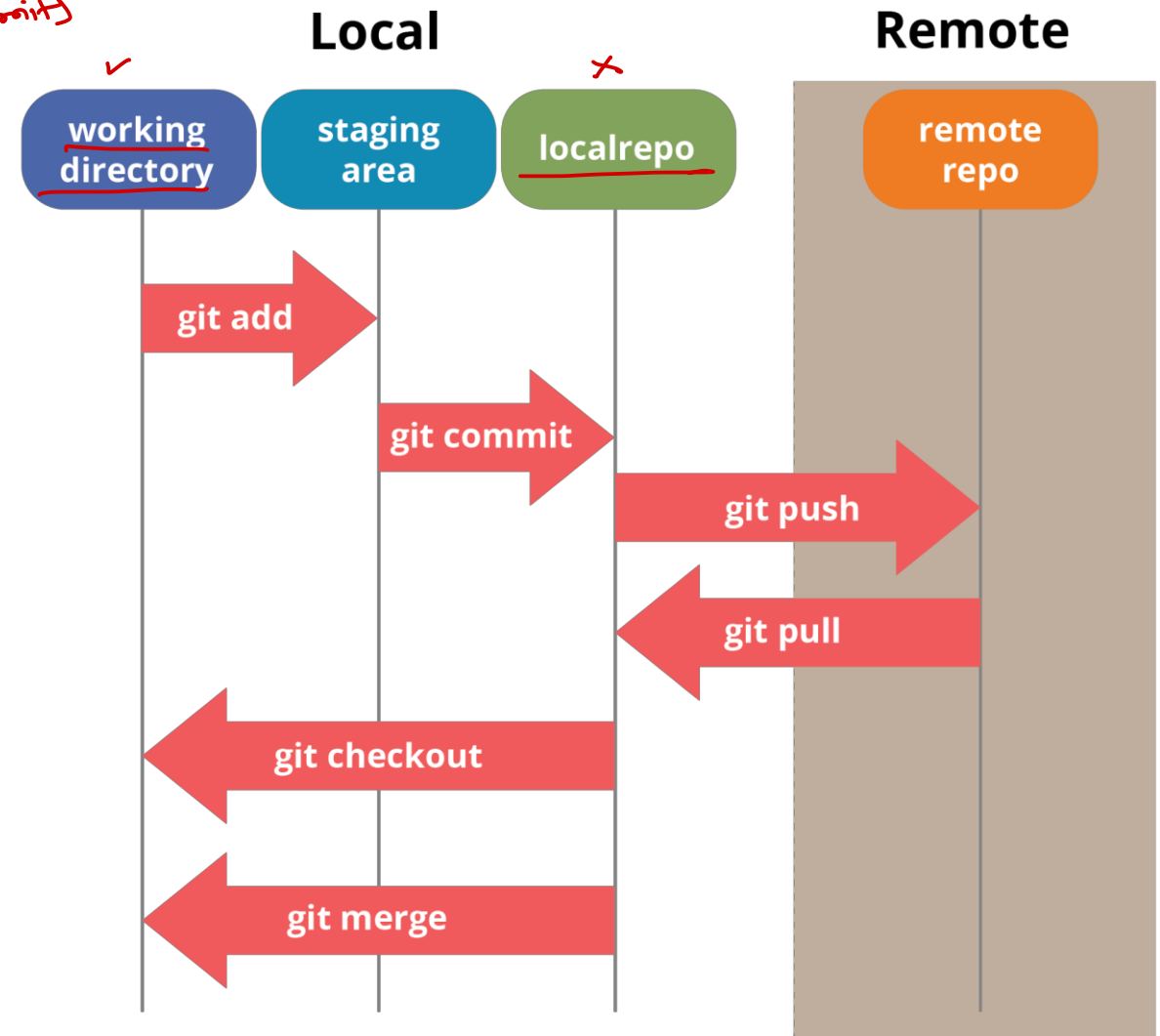
- Local vs Remote repository

- Remote repository operations

- clone, pull, push.

*Handwritten notes:*  
•  $\frac{\Delta C}{\Delta t} / \frac{\Delta h}{\Delta t}$   
• \*.java  
• \*.py  
• delta  
• dev time  
• id (commit)  
• loc

*Handwritten note:*  
• directory & subdirectories & file.



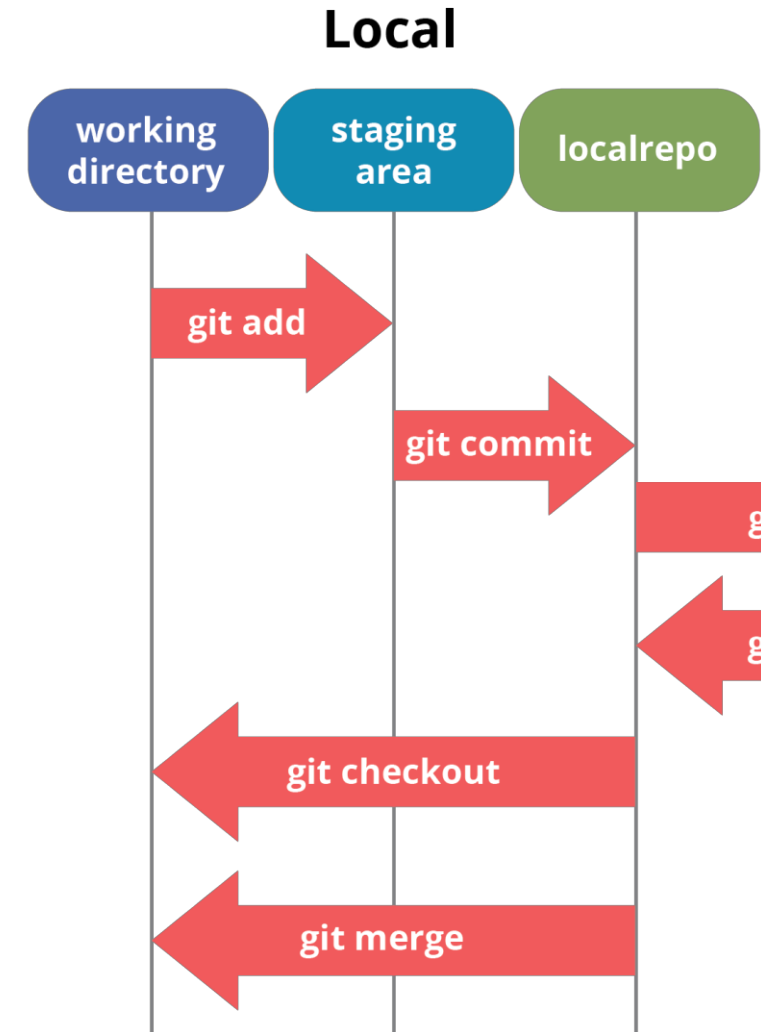
# Git installation & setup

- On Ubuntu
  - `sudo apt-get install git`
  - List global settings
    - `git config --global --list`
  - Set up global properties
    - ✓ `git config --global user.name <your name>`
    - ✓ `git config --global user.email <your email>`
    - `git config --global core.editor <editor app>`
  - GIT user details are associated with each commit done by the user.
- On Windows
  - Download and install GIT.
    - <https://git-scm.com/downloads>
  - Installed components
    - GIT bash
    - git-gui + gitk
  - GIT Bash
    - git command
    - bash commands
    - vim editor
- In editor/IDE
  - All leading IDEs have GIT support.
  - VS Code, Eclipse, ...



# GIT commands

- terminal> git init
- terminal> git status
- terminal> git add .
- terminal> git commit -m "message"
- terminal> git diff  
(track changes that are not staged)
- terminal> git checkout file  
(discard changes & get last committed version)



# .gitignore

---

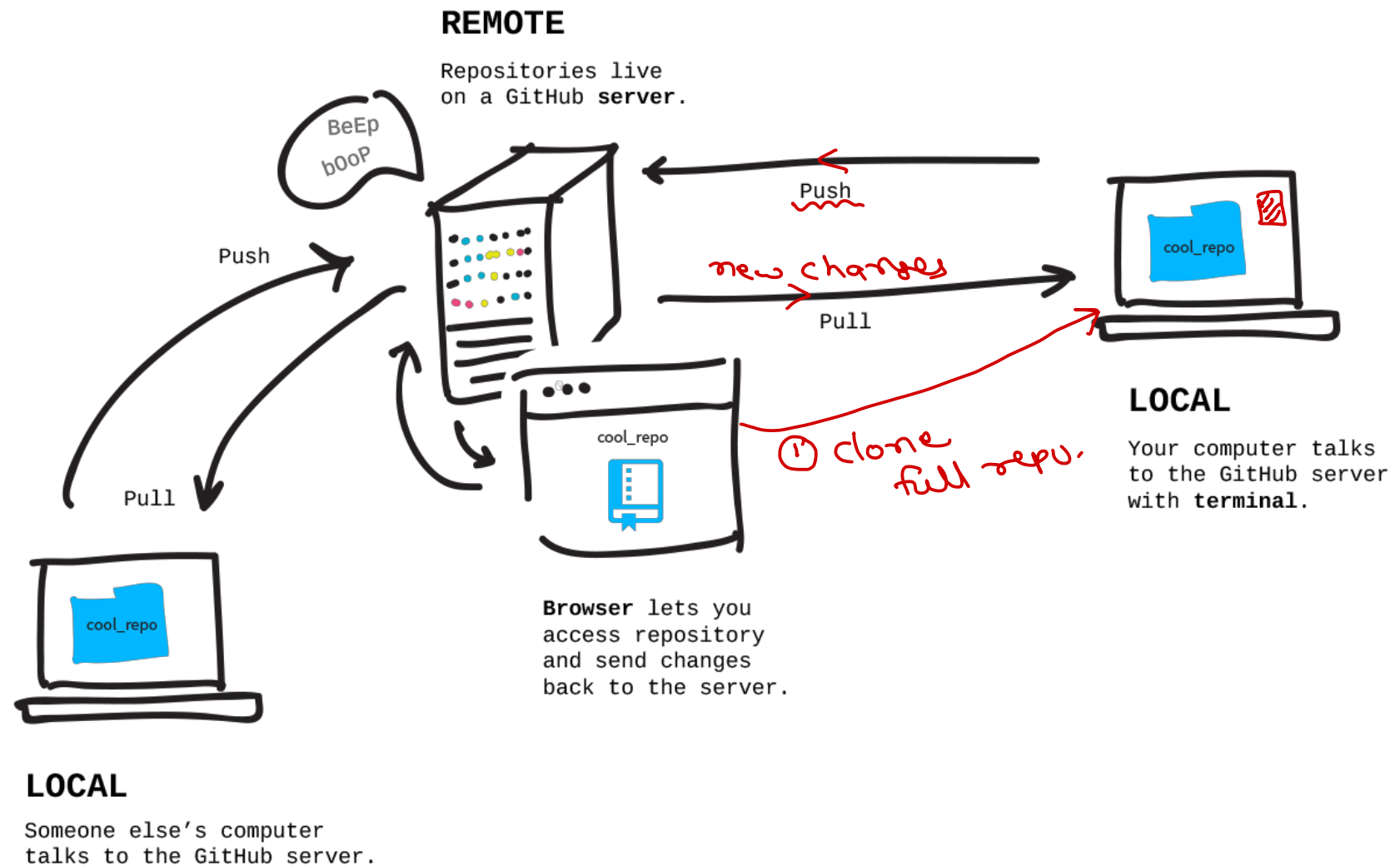
- List directories or files to be ignored for git repository.
- Used to ensure that binaries, IDE metadata files and other undesired files are not maintained in git repository.





# GIT Remote repository

- To maintain code repository at centralized location (for code sharing).
- Can be in intranet or internet.
- Popular vendors
  - github.com
  - gitlab.com
  - bitbucket.org



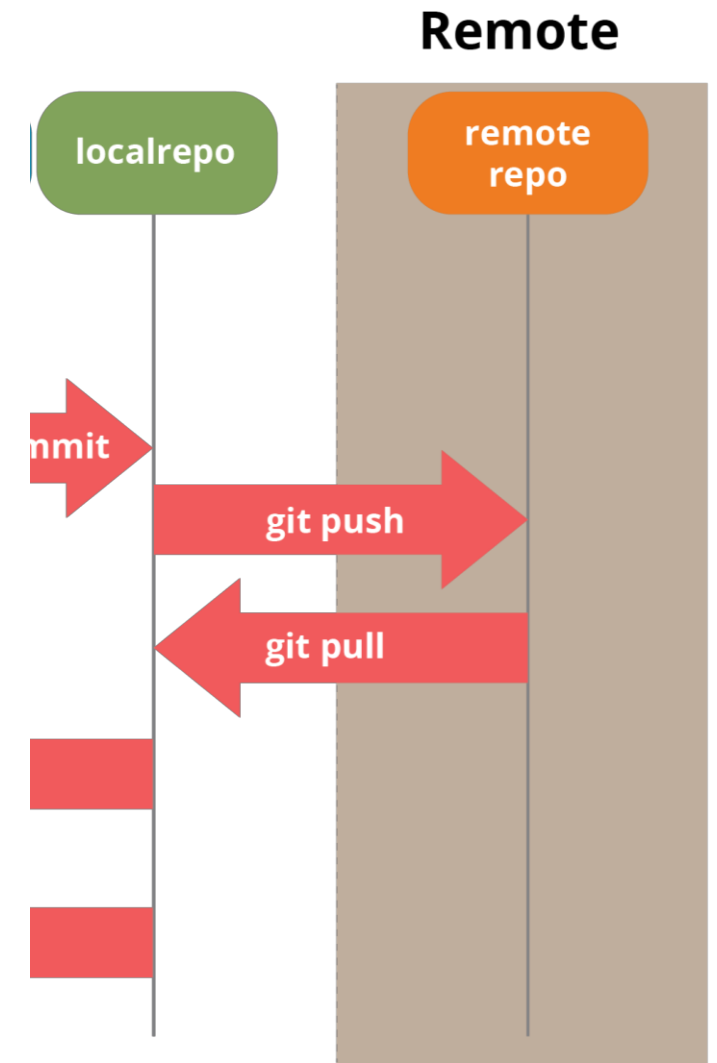
# GIT commands

- `git remote add origin <remote url>`

- `git clone <url>`

- `git push origin <branch>`
  - master (older projects)
  - main (newer projects)

- `git pull`



# GIT workflow ← Most Commonly Followed.

- Create project on gitlab.
- Clone repository on local machine.
- Add/modify code locally.
- Commit code in local repository.
- Push code to gitlab repository.
- Other developers can pull your code.



# GitLab

- **gitlab** project repository

- Members
- Invite members (member id/email id)
- Choose role (Developer/Maintainer)

for project team members } Can read & write.  
for reviewers or evaluators } Can only read

- gitlab protected branches

- master branch is protected  
*main*

- gitlab roles

- Maintainer
- Developer

- Assignment evaluation

→ make logical commits as mentioned in assign doc.  
→ add lab mentor as Developer in your project (as per coordinator instruction).





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

