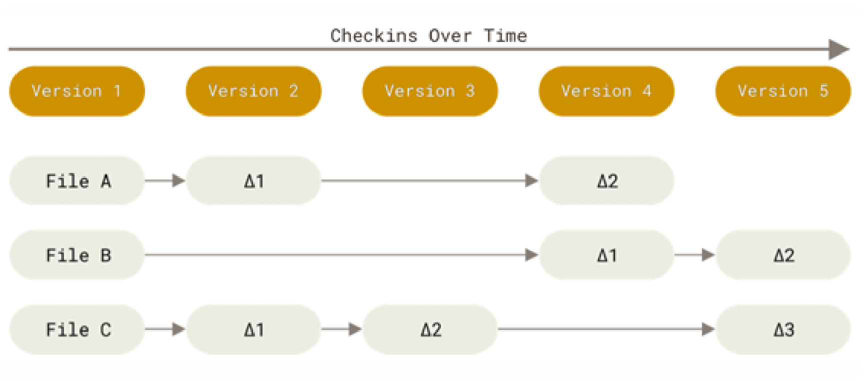
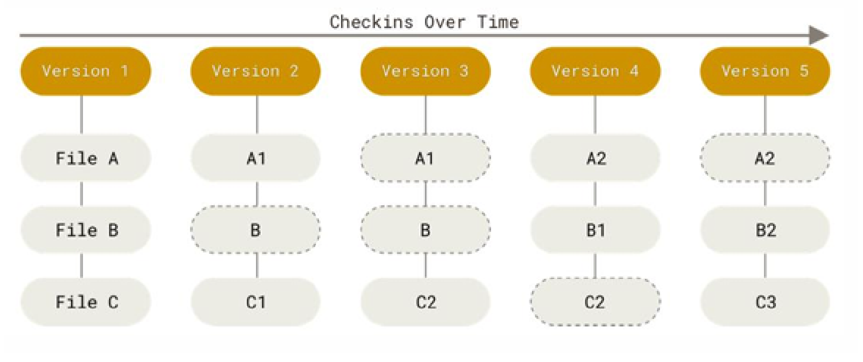
Git

▼ Version control system

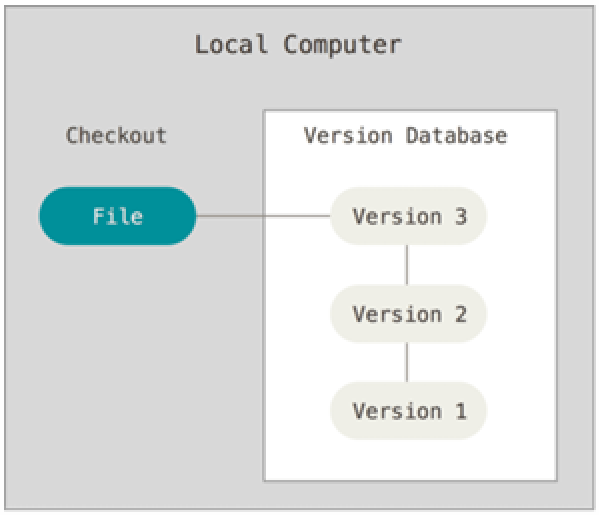
- Data store types
    - Storing data as changes to the base version



    - Storing data as sanpshots ← Git

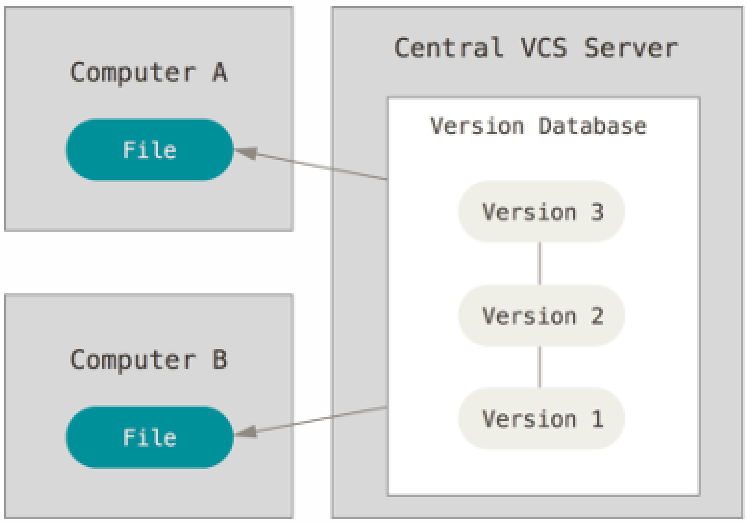


- Version control types
    - Local



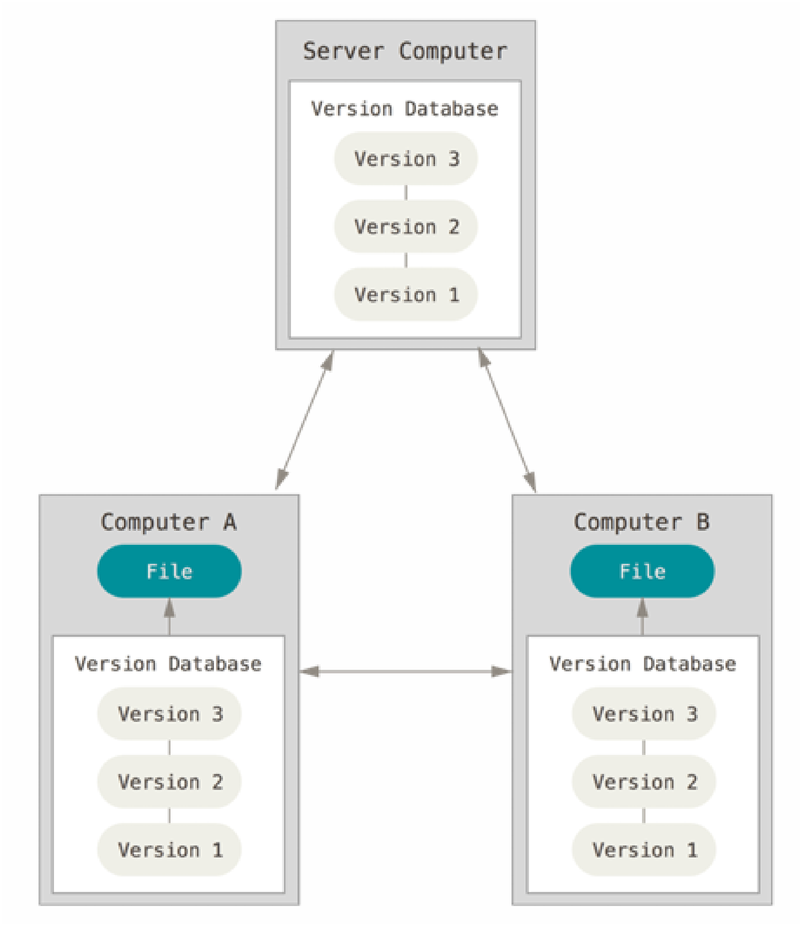
Only in my personal computer, so not good at team project

- Centralized



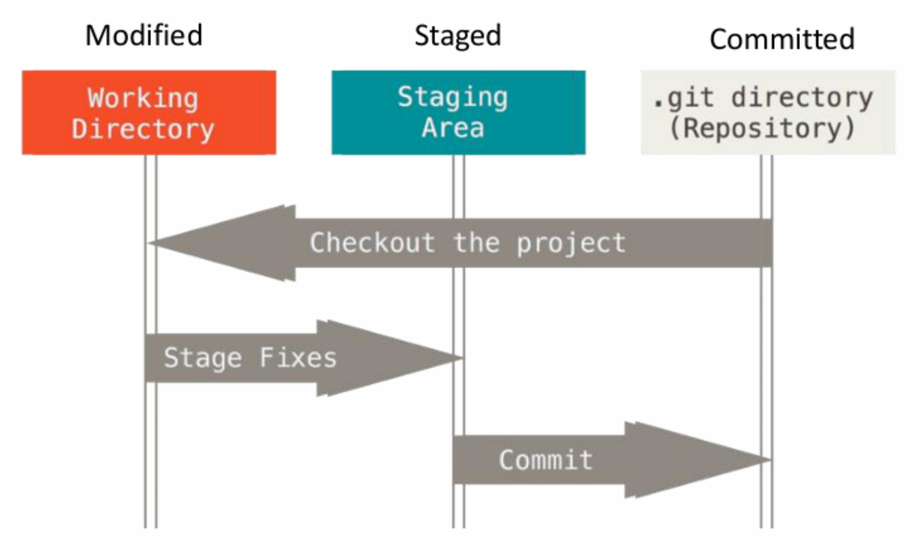
Storing data in server computer. It is able to do team project, but depends on reliability of server computer

- Distributed ← Git



To avoid risk of centralized type, we also have data in personal computer

- States



  - Modified

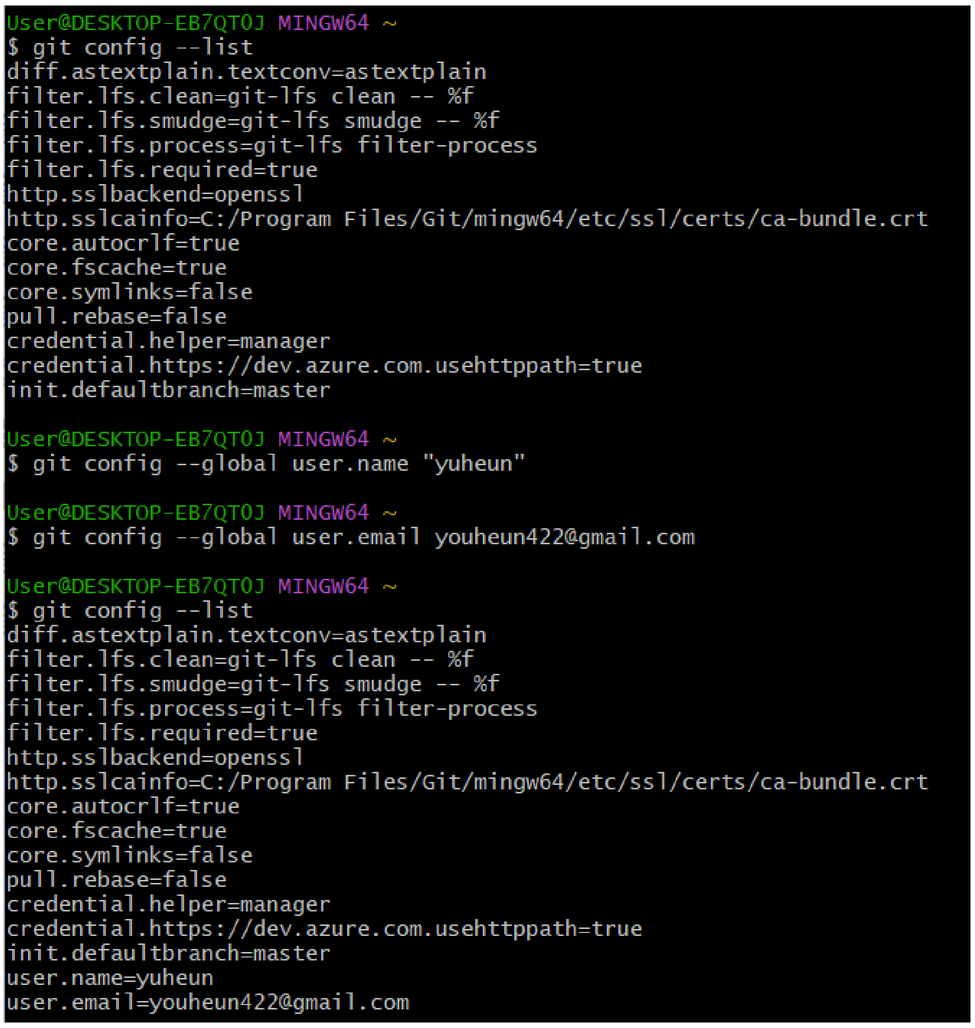
- Staged
- Committed : snapshot finished (stored)

▼ Lab

- Git config setup

```
file: /etc/gitcongig # system lv
file: ~/.config/git/config # user(global) lv
file: .git/gitcongig # local lv
```

```
$ git config --list # info check
```



```
User@DESKTOP-EB7QT0J MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master

User@DESKTOP-EB7QT0J MINGW64 ~
$ git config --global user.name "yuheun"

User@DESKTOP-EB7QT0J MINGW64 ~
$ git config --global user.email youheun422@gmail.com

User@DESKTOP-EB7QT0J MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=yuheun
user.email=youheun422@gmail.com
```

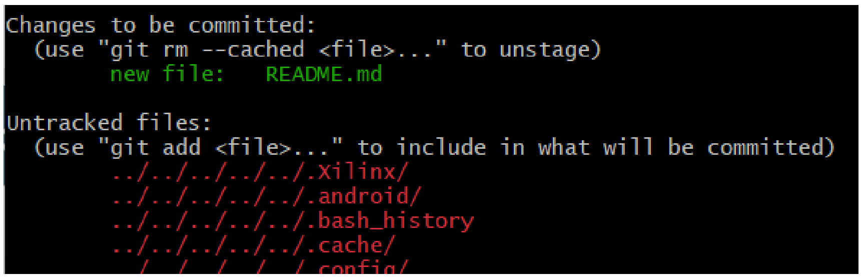
> check changes!

- Init a repository

```
$ git init
$ git status # check the repository status
```

- Make file to be staged

```
$ git add README.md
```



```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../../../../../../.Xilinx/
        ../../../../../../.android/
        ../../../../../../.bash_history
        ../../../../../../.cache/
        ../../../../../../.config/
```

```
$ git add . # add all files to be stage
```

- Make file to be unstaged

```
$ git rm --cached README.md
```

- Ignore a file
  - *.py

    # ignore all .py files
  - !lib.py

    # do track lib.py even though ignored .py files above
  - /TODO

    # ignore TODO file only in the curr dir
  - build/

    # ignore all files in any dir named build
  - doc/*.txt

    # ignore doc/notes.txt but not doc/server/arch.txt
  - doc/**/*.pdf

    # ignore all .pdf in the doc/ dir and any of its subdir

- Commit

  ```
  $ git commit -m "commit msg"
  $ git log
  ```

- Change branch name

  ```
  $ git branch
  $ git branch -m master main
  ```