Distributed Version Control Systems – Git, Mercurial, Bazaar or Darcs

Local, centralized and distributed version control systems

Git – free, open source, fast, scalable, cheap branching/merging

GitKraken – best GUI for gits

https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

pass: GHVichingo39!

Check out visual studio code as an editor

to config the git account locally:

git config --global user.name "vgi-nikolay-aleksiev

git config --global user.email "nikolaya.aleksiev@vgipartners.com

git config –global code.editor “code --wait” # code is an editor added to the system variable

git config –global core.editor notepad

git config –global -e # to configure all the settings in a text file (opens in default editor)

git config –global core.autocrlf true # true for Windows, input for mac or linux

git config –global diff.tool vscode # to set the diff tool (Kdiff3 or P4Merge or vscode

git config –global difftool.vscode.cmd “code –wait –diff $LOCAL $REMOTE”

cd ../ to folder

"git init" # to initialize a git:

"git clone <https://github.com/vgi-nikolay-aleksiev/Test>" # to clone a directory:

ssh-keygen -t rsa -b 4096 # Ssh keys

ssh-add

**IN GITBASH (navigate to the git folder)**

After changes set up the staging area (index):

1. **git status** # to test status of working directory on staging area

git status -s # brief status (left col staging area, right col working dir)

git diff # shows difference between working dir and staging area

git difftool # can use diff tools – dir vs staged

git difftool --staged # only changes to the staged area

1. git add firstfile.txt or \*.txt or **.** # to add new files to git:

git add secondfile.txt # would delete file from staging area in not in folder

git rm file1.txt # removes file/s from working directory and staging area

git rm –cashed file1.txt # removes file/s from staging area

git mv file1.txt file2.txt # renames file1 to file2

.gitignore file and add files and directories # add files and directories to get ignored when committing

# github.com/github/gitignore gives you language spec .gitignore

1. git commit -m “initial commit” # to commit all files

git commit # can specify the comments in a file

1. git log –oneline --reverse # logs of all commits

git show d601b90 # to dig into a specific commit

git ls-tree d601b90 # to show all blobs (files), trees (directories), tags in a commit

git show-ref # to get commits, blobs (files), trees (directories), tags

git restore –source d601b90 file2.txt # restore a deleted file from previous commit

1. git remote add origin ‘the url.git’ # add origin where the data gets pushed to
2. profile -> settings -> developer settings -> PAT
3. Credential manager -> generic credentials -. Git:https://github.com
4. Git config –global credential.helper wincred
5. git push -u origin main # to push a commit: