To connect to your github account from the command line

Install credential manager (<https://github.com/cli/cli/blob/trunk/docs/install_linux.md>)

*curl -fsSL https://cli.github.com/packages/githubcli-archive-keyring.gpg | sudo dd of=/usr/share/keyrings/githubcli-archive-keyring.gpg*

*echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/githubcli-archive-keyring.gpg] https://cli.github.com/packages stable main" | sudo tee /etc/apt/sources.list.d/github-cli.list > /dev/null*

*sudo apt update*

*sudo apt install gh*

go to your github->developer settings->personal access tokens

generate a new access token and copy it

*gh auth login*

select https as access method follow the prompts. This will mean you can push to github without needing a password each time.

Might need to instead do:

git config credential.helper store

Then when you push your credentials are stored after you first enter them

Creating a repo and connecting it to github

Either:

Create a local repo with git init

Create a repo on github

*git remote add* origin <repo URL>

Git pull origin main

Or

Create a local repo with git init

*git checkout -b main* (create a main branch to work in to match the default branch on github, default branch in git is master)

make changes

add changes

Commit

Create a repo on github (with nothing in it)

*git remote add origin* <repo address>

*git push -u origin* <local branch name>

e.g. *git push -u origin main*

Or

*git push --set-upstream* <local name of repo> <local branch name> (then just git push for future commits)

A simpler way to work on a local copy of a remote repo is to use git clone. This creates a local copy of the repo in its own directory, where you can work on the files, and git add/commit/push as needed (no need for the above set up commands).

Some useful git commands

git log to list commits

git reset –hard <commit ID> to rollback to a particular ID (remove hard to keep the reverted changes locally)

git branch - see what branches there are, the one you are on is marked \*

git switch <branch name>

git status - get current state of repo

Diff and merge for conflict resolution using Meld

Install meld

*git config --global diff.tool meld*

*git difftool* <other branch name>

*git config --global merge.tool meld*

*git merge*

#conflict!

*git mergetool*

Decide how to resolve the conflict, then save all changes, and exit meld. Then commit the new merged files