Deren Bozer
COSC-381 MW 11:00
Winter 2023

Git Cheatsheet

Setting up Git

Configure the username & password

- Set your global user name
  - Command: `git config --global user.name "<your name>"`
  - Example: `git config --global user.name "DerenB"`
- Set your global emails
  - Command: `git config --global user.email "<email>"`
  - Example: `git config --global user.email "dbozerprogramming@gmail.com"`

Configure the Editor

- VSCode: `git config --global core.editor "code --wait"`

Generate a SSH key

- `ssh-keygen -t ed25519 -C "<your email>"`
  - Terminal will show: `Generating public/private ed25519 key pair.`
- Terminals Prompts for a file location to save the key:
  - EC2: `/home/ec2-user/.ssh/github_ssh_key`
  - MAC: `<path-to-your-location>/github_ssh_key`
  - Windows: `<path-to-your-location>/github_ssh_key`
- Press Enter twice to skip using a password
- Add SSH key to ssh-agent
  - Start the ssh-agent: `eval "$(ssh-agent -s)"`
  - Add the new key: `ssh-add ~/.ssh/github_ssh_key`
- Add the key to your GitHub account
  - Go to the Github website > account > settings

    - SSH and GPG keys
    - Copy the key located in `github_ssh_key.pub`
- Test the Connection:
    - `ssh -T git@github.com`

Repo Creation

- Clone Repository
    - Command: `git clone repo1 repo2`

Creating and viewing Repo

- Initialize a directory into a Git repository
    - Command: `git init`
- Change the name of the Main/Master branch
    - Command: `git branch -m <newName>`
    - Example: `git branch -m main`
    - Changes the name from the default "master" to "main"
- Show the status of the repo
    - Command: `git status`
- Show the history of a repo
    - Command: `git log`

Connecting to a repo from the Command Line

- `echo "Anything" > readme.md`
- `git init`
- `git add readme.md`
- `git commit -m "First Commit"`
- `git branch -M main`
- `git remote add origin get@github.com:DerenB/repoName`
- `git push -u origin main`

Adding to a Repo

- Add files to be committed
    - Command: `git add <FileName>`
    - Example: `git add readme.md`

- Check the un-staged changes
  - Command: `git diff <FileName>`
  - Example: `git diff readme.md`
- Add all changes in the staging area into one commit
  - Command: `git commit -m "<Commit Message>"`
  - Example: `git commit -m "My First Commit"`

Branches

- Create new branch
  - Command: `git branch <branch_name>`
  - Example: `git branch BugFixes`
- List Branches in Repo
  - Command: `git branch`
- Display Current branch
  - Command: `git branch --show-current`
- Switch to Branch
  - Command: `git checkout <branch_name>`
  - Example: `git checkout BugFixes`
- Delete Branch (safe)
  - Command: `git branch -d <branch_name>`
  - Example: `git branch -d BugFixes`
- Delete Branch (force)
  - Command: `git branch -D <branch_name>`
  - Example: `git branch -D BugFixes`
- Rename Branch
  - Command: `git branch -m <branch_name>`
  - Example: `git branch -m MyNewBugFixes`
- List all remote branches
  - Command: `git branch -a`

Stash & Pop

- Stash branch changes for later
  - Command: `git stash`
- Stash untracked files
  - Command: `git stash --include-untracked`

    - Command: `git stash -u`
- List Stashes
    - Command: `git stash list`
- Show the latest Stash
    - Command: `git stash show`
- Pop stash changes back into the working branch and remove from stash
    - Command: `git stash pop`

Remote

- List Remotes
    - Command: `git remote`
- Show Origin
    - Command: `git remote show origin`
- Remove remote
    - Command: `git remote rm <remote_name>`
- Push remote to new branch
    - Command: `git push -u origin old_branch:new_branch`

Fetch

- Fetch branch
    - Command: `git fetch ../<repo> <branch>`
- Merge Origin
    - Command: `git merge origin/<branch>`