2 Git Commands and User Interface

Arni Magnusson

SPC Git/GitHub Workshop
Noumea, 13 April 2022

Overview

Pacific Community
Communauté du Pacifique

First steps *clone, status, diff*

Submitting changes *add, commit, push*

Refresh and document *pull, log, tag*

Undo things *rm, checkout, clean, reset*

User interface *command line, rstudio, web browser, other*

Pacific Community
Communauté du Pacifique

First Steps

First Steps

clone Download a copy of remote repo to your computer

status See files you have changed

diff See lines you have changed

.git is a hidden folder containing a database of tracked changes

Pacific Community
Communauté du Pacifique

Submitting Changes

Submitting Changes

add Specify files to commit

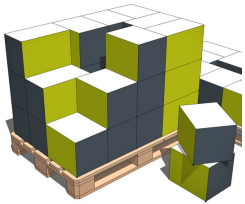
commit Describe what was changed `git commit -m "My commit message"`

push Upload to remote repo

3-step combo move

Submitting Changes

add



select boxes
that go together

→

commit

wrap and label

→

push

send it off

Commit Messages

Pacific Community
Communauté du Pacifique

Tips for writing commit messages

- Relatively short *$\sim$ 60 char*
- Describe purpose *or specific things that were changed*
- Present tense *often starting with a verb like 'Add', 'Improve', 'Document', ...*

Examples

- MFCL
- skj22
- yft-2017

Commit Messages

Pacific Community
Communauté du Pacifique

Refresh and Document

Refresh and Document

Pacific Community
Communauté du Pacifique

pull Refresh local repo *download latest commits from remote repo*

log View commit messages *can also view on github.com*

tag Mark significant commit with a nickname *example: 2.0.0*

SHA Hash

Every Git commit has a unique identifier

This identifier (SHA hash) is 40 characters long

random combination of `0123456789abcdef`

It's enough to use the first 7 characters

For example, the first commit of this workshop's GitHub repo:

```
    e15ccce7ad92c6ae5a67b707816add8da25d4cfa
= e15ccce
```

We use tags to give commits meaningful nicknames

Pacific Community
Communauté du Pacifique

Undoing Things

Undoing Things

Pacific Community
Communauté du Pacifique

rm Remove file from repo *it will still be 'remembered' in history*

checkout Undo local changes `git checkout -- file`

(also) Select branch

clean Remove files that are not part of the repo *example: MFCL output files*

reset Rewind to a previous snapshot of the project *pull to go back to the 'present'*

(also) Un-add files

Pacific Community
Communauté du Pacifique

User Interface

User Interface

Pacific Community
Communauté du Pacifique

Command line Works on all computers, full Git functionality

especially practical if you know the basic shell commands

RStudio Convenient if you're already using RStudio

small subset of Git commands, noticeable lag, creates 'garbage' files

Web browser Commit history, files and lines changed, GitHub features

Other Most editors have Git support, also many dedicated Git clients

view current changes (not yet committed)

User Interface (cont)

Pacific Community
Communauté du Pacifique

Arni's shell scripts and aliases

add, br, clone, co, commit, di, log, merge, pull, push, reset, show, st, stash, tag *shorthand*

c *clone and rename OFP-SAM repo*

commits *that are not yet pushed*

eg *Emacs Magit*

files *that were changed in a given commit*

lfh `log-full | head`

shortlog *show commits by author*

tag-delete *remote repo*

url *show remote repo location*

br-full, commits-full, files-full, log-full, shortlog-full, show-full, tag-full

Summary

First steps *clone, status, diff*

Submitting changes *add, commit, push*

Refresh and document *pull, log, tag*

Undo things *rm, checkout, clean, reset*

User interface *command line, rstudio, web browser, other*