

Git worksheet

Carey Pridgeon

November 17, 2016

1 intro

- In this worksheet you will cover the basics of using Git by using gitlab.

2 git Commands

- **git clone [url]**
 - Clone a remote repository.
- **git add [filename]**
 - Add a specific file (need especially for dot prefixed, and thus hidden, files)
- **git add -all**
 - Add all files not specifically excluded by the .gitignore file.
- **git commit -m "message"**
 - Commit any changes to the local repository
- **git push**
 - update the remote repository with local changes.
- **git config -global user.name "John Doe"**
 - **git config -global user.email johndoe@example.com**
- set up git locally so it knows the ID to use on commits/pushes
- **git config -global core.editor emacs**
 - If you want to change the default editor, use this
- **git config -list**
 - display your git config

- `git tag -a [tag_name] -m "tag text"`
 - create a tag with detail text
- `git checkout -b [branchname]`
 - Create a new branch and switch to it
- `git push -u origin [branchname]`
 - Push your local new branch and its changes to gitlab.
- `git checkout [branchname]`
 - Switch to an existing branch

3 Gitlab runner setup

- Gitlab provides a CI service that you can use for compiling and testing.

- To use a shared runner you need to set it up by creating a runner script, which is written in yaml. A script starts like this:

Heading

```
sectionname:
- command
- sequence
```

- The command sequence is the same as a sequence of bash commands. So `cd` would change directory, `make` runs a makefile.
- Look at the following `.gitlab-ci.yml` file that can compile a c++ project.

mopdev:

```
script:
- cd [source_folder_name]
- make clean
- make
- [./executableName]
```

- <https://gitlab.com/carey.pridgeon/mopfile>

4 Tasks

4.1 Playing with the repository

- Create an account on gitlab using your real name.
- Import your public ssh key from nostromo to your gitlab account. screenshot this is when complete. Add the screenshot and your `id_rsa.pub` in your report. It, and your account will be deleted at the end of the module, so this is not a security issue.
- In this account, fork this existing repository:
- https://gitlab.com/carey.pridgeon/git_task
- Remove the connection to my repository in the settings for your fork.
- Set up your local git id
- Clone this repository into nostromo.
- In this repository, add the file `.gitignore`.
- In this file add `.o`. This will prevent git tracking object files (compilation artefacts)
- Add this file to the repository
- Push your changes upstream to the remote repository
- Create a new file `README` and write something in it
- Add this file to the local repository with the commit message *added a readme*, then push it upstream.
- Create a tag called 'before_branch', push it upstream
- Make a new branch called "dev_branch", switch to it, make some changes (anything, just edit the readme if you like) then push them upstream to the new branch.
- Merge the branch with your master branch on gitlab (not the original from my repository), deleting the new branch in the process, and switch back to using master branch locally.

4.2 Set up CI

- Create the file `.gitlab-ci.yml`, setting it up to compile, then run the program
- Push it to the remote repository to trigger a build

5 Evidence

- Submit a screenshot of the repository history page. The full text output from the build, plus the the console history covering this worksheet in a single document.