Title.

Name: ...

18-04-2018

1 Introduction

1.1 What is github

Github and Gitlab can be used to store projects containing code (or any file) on a server at Github. You can:

1. Log in to the service gitlab and type/upload your files in the browser.
2. install the software named Git and:
   - (a) use it from the command line.
   - (b) use it like a "normal windows" program with a "normal user interface".

1.2 Content of this manual

1. Using Gitlab with the TUD account.
2. Some minor steps required to be able to connect from commandline to the service (generating an SSH key).
3. ***UsingthecommandlineinterfacetoactuallyuseGitlaband/orGithub***

2 topics are discussed that are not solved here:

1. installing the program Git, which is essentially github on your pc (it also works for gitlab.
   - (a) You can download it here:
   - (b) `https://git-scm.com/downloads`
   - (c) and keep all settings at standard unless you want to generate your ssh key with Putty, then look here first:
   - (d) `https://docs.joyent.com/public-cloud/getting-started/ssh-keys/generating-an-ssh-key-manually/manually-generating-your-ssh-key-in-windows#Git-Bash`
2. Building your ssh key with Putty in stead of Git (bash). (bash is a different form of the programming language shell, it is also the interaction language for a lot of different command prompts in Linux named e.g.: Terminal).(I oversimplified that, possibly wrong.)

2 Getting started with Gitlab

2.1 Gitlab SSH key

For github creating an SSH key is not really necessary as far as I am aware, but Gitlab showed a message I would not be able to download/clone the repository if I did not have an SSH key. This is how to generate your own public and private SSH key:

1. First lets make something explicit:
2. There is a difference between the gitlab account given by Ewi Tudelft and a general gitlab account
3. https://gitlab.ewi.tudelft.nl/
4. gitlab.com
5. They both use different passwords and different user names

Now for the SSH key generation:

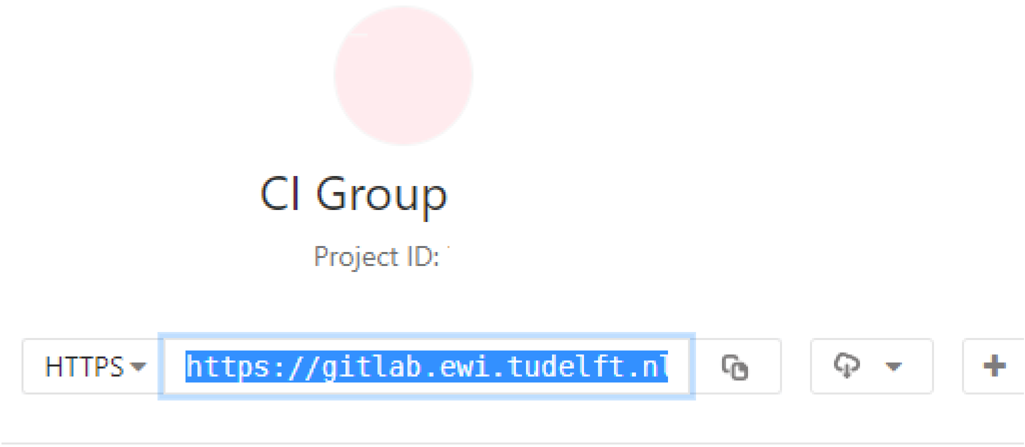
1. Source: `https://docs.joyent.com/public-cloud/getting-started/ssh-keys/generating-an-ssh-key-manually/manually-generating-your-ssh-key-in-windows#Git-Bash`
2. Assuming you have Git with bash installed in windows:
3. cd `"C:\Users\a\Documents\fun\security"`
4. `"C:\ProgramFiles\Git\bin\bash.exe"`
5. mkdir .ssh
6. ssh-keygen.exe
7. The output then looks something like:
8. Generating public/private rsa key pair
9. Your private file is the file named "keygithub" without an extension
10. Your public file is the file named "keygithub.pub"
11. Next you open the file "keygithub.pub" with notepad or sublime Inside you find your sha/rsa key starting with:
12. ssh-rsa <long code> <a your computer name>
13. You can change the computer name, it is just for you so you know which key it is.
14. If you enter it somewhere, somewhere can use that computer name to name the key, so change it if you don't want the name of your computer public.
15. You can enter that key in: "gitlab>settings>ssh"

2.2 Start using Gitlab with TUD

Now this is where you actually start using the software for what it is intended to do, provide simple and fluent software creation in a team environment. If you want to edit the files locally, you have to start with putting the project on your local computer. Also known as downloading the project, also known as cloning the project. This describes the first time downloading a project:

2.2.1 "Cloning"/downloading a "repository"/folder/project

1. To download a repository (that is what the project/folder on the gitlab server is called):
   (a) Open an internet browser, if you want to download a University repository of the EWI faculty, go to https://gitlab.ewi.tudelft.nl/ (not `gitlab.com`).
   (b) Login with your tudelft email (and password)
      i. You should have received an invitation to activate your account from the Brightspace course that gave you the university gitlab account in the first place.
   (c) After logging in browse to the course on gitlab, then browse to your group, then browse to your "project/repository" then you see blabla "SSH" then do the stuff from the previous section.
   (d) if there is no red header saying stuff about ssh you can select the link to download(clone) the repository as shown in fig. 1



HTTPS https://gitlab.ewi.tudelft.nl

Figure 1: This is where you copy the link from. Set to Https for starters.

(e) Set the link type to ”HTTPS” and copy the link shown. (refered to as <repository link>)

(f) Make a local (local means on your pc) folder

(g) E.g. ”C:/test folder”

(h) open command prompt by going to start>type cmd>click on command prompt.

(i) cd ”C:/test folder”

(j) cd git clone <repository link>

(k) now check in the folder if it is there.

3 Gitlab/Github commands

There are options that deal with branching(different versions of the same project), merging (chosing 1 of the two versions to become the master(main) branch(version) of the project) and conflicts between different different versions. I will not go into that and just stick to the simple stuff:

3.1 Meaning of the commands:

1. git pull: ”Check if my local ”copy” of the project is up to date with the most recent ”copy” of the project in the Github/Gitlab server. If not, it downloads and/or changes the missing/outdated files.

2. git add: ”Indicate which folder or file, (relative to your local Git folder/repository) you want to upload to the Gitlab/hub server.

3. Git commit: Add a comment so others know what you changed

4. Git push: Force your changes upon the server. (uploads your file and overwrites the old ones).

3.2 Usage of the commands:

1. To use them, open cmd and browse to the folder.

2. the commands are: ommands for adding your python code to github:

3. git pull

4. git add src/name of file0.py

5. git add src/name of file1.py

6. git commit -m ”This is the description of your update/comment”

7. (if you are requested to to add credentials do it with the lines they provide, unless its a pop up box, then just fill in your tudelft email and the gitlab password you set on that account.)

8. git push origin master

1. comments:

2. (updates your local files, appearently without overwriting your new ones with the old ones, as long as someone else did not work and push in your new file.)

3. (For all the files you want to update, everytime you make changes, so not just the first time you create that file)

4. *for us the path of 1 could for example become,:

5. git add `src\models\deployment\deployment.py`

6. Note, that the absolute path, does not work:

7. WRONG: - 1. git add `C:\some_folder\src\models\deployment\deployment.py`

4 Troubleshooting

4.1 Credential/password troubles

1. If you are prompted for the EWI gitlab pasword over and over

2. do not go to github.com to click forgot password to enter your TUDelft email,

3. Go to https://gitlab.ewi.tudelft.nl/ and do it there.

4. Also you might have not set the Git clone link to HTTPS, do that before you copy it, as shown in fig. 1

Now you know how to use and download projects from github, you can try to find a fun (automation) script from some repository to download and use, it can open a whole new world!

References