Gitlab implementation process

This process allows you to clone a project or a project folder from the Gitlab app on NAS server, to your computer.

If you have questions or issues with some steps, please refer to :

HAJJI Ismail ("Hajji Ismail" <Ismail.Hajji@curie.fr>)
(or) LAGARRIGUE Lilian ("Lagarrigue Lilian" <lilian.lagarrigue@curie.fr>)

Step 1 : Download and install Git on your PC

https://github.com/git-for-windows/git/releases/download/v2.35.2.windows.1/Git-2.35.2-64-bit.exe

For the following steps, you can refer to the official gitlab protocol online on this website :

https://www.instructables.com/Gitlab-CE-on-QNAP-NAS-Installation-Basic-Usage/

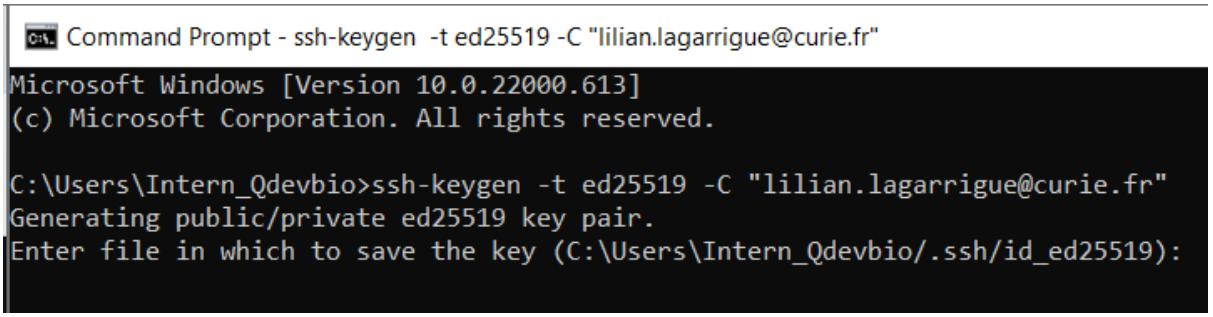
Step 2 : Open the command prompt

(“cmd” in windows research bar) - the admin mode isn’t necessary

Step 3 : Generate the public and private ssh keys

Type this command by replacing the email by yours:

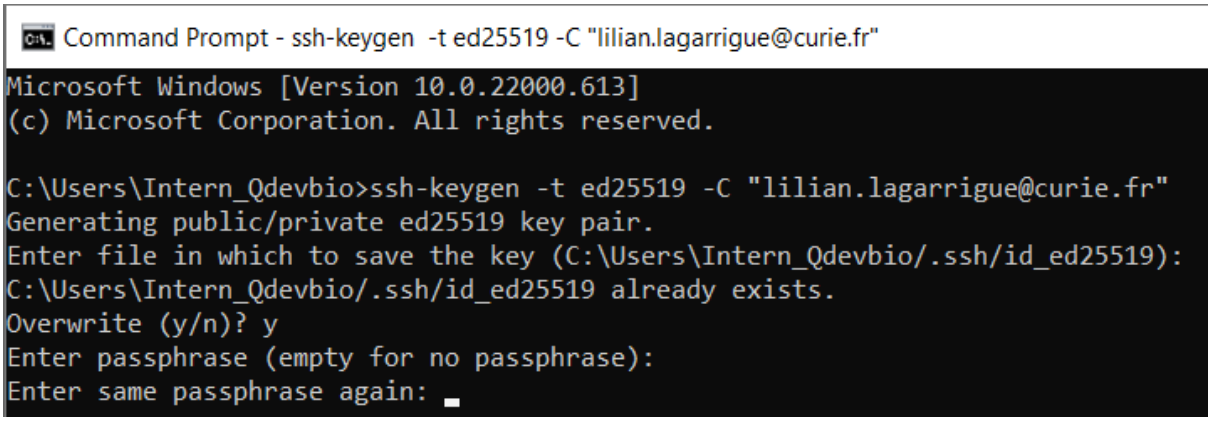
>>> ssh-keygen -t ed25519 -C "email@curie.fr"



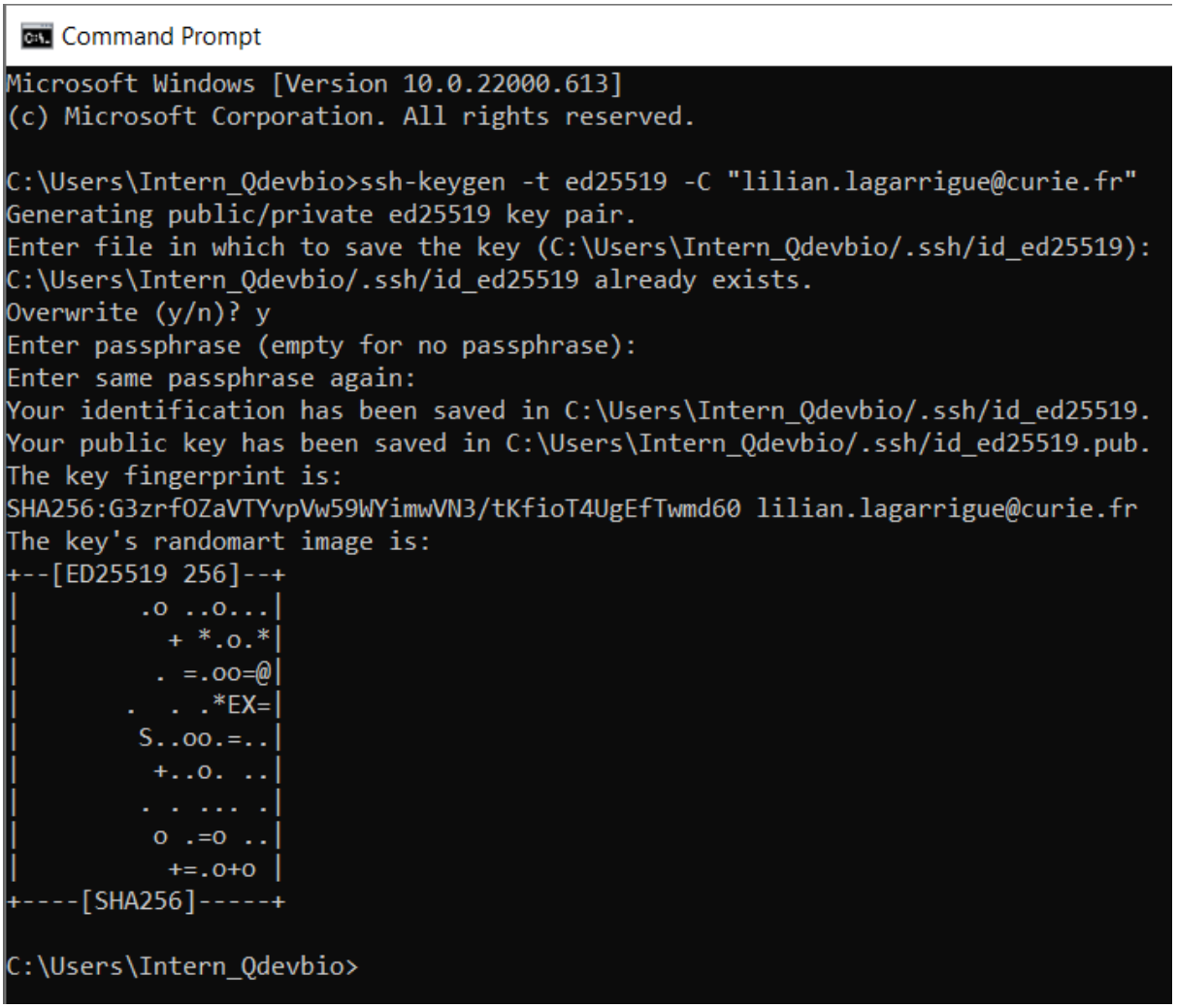
Press “ Enter ” for saving the key in the file which will be named as “ id_xxxxx ” (“id_ed25519” in this example).

Step 4 : Generate the public and private ssh keys

4.1 - If the public and private key files aren't already created, enter a new passphrase twice



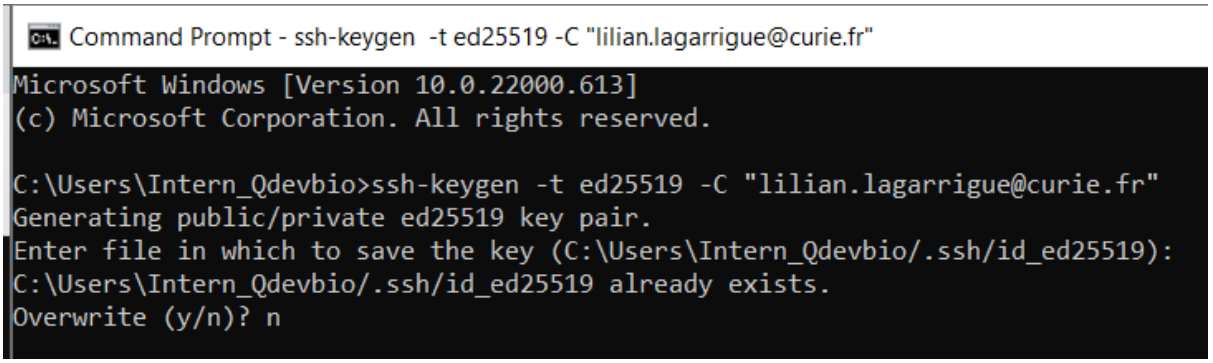
Enter the passphrase that you will have to enter each time you want to import/export a file or a folder from gitlab (this password protects the private key associated with your PC).



As you see above, you should see something that corresponds to your key's randomart image.

4.2 - If the public and private key files are already created, it is not necessary to overwrite the ssh private key file => no overwriting:

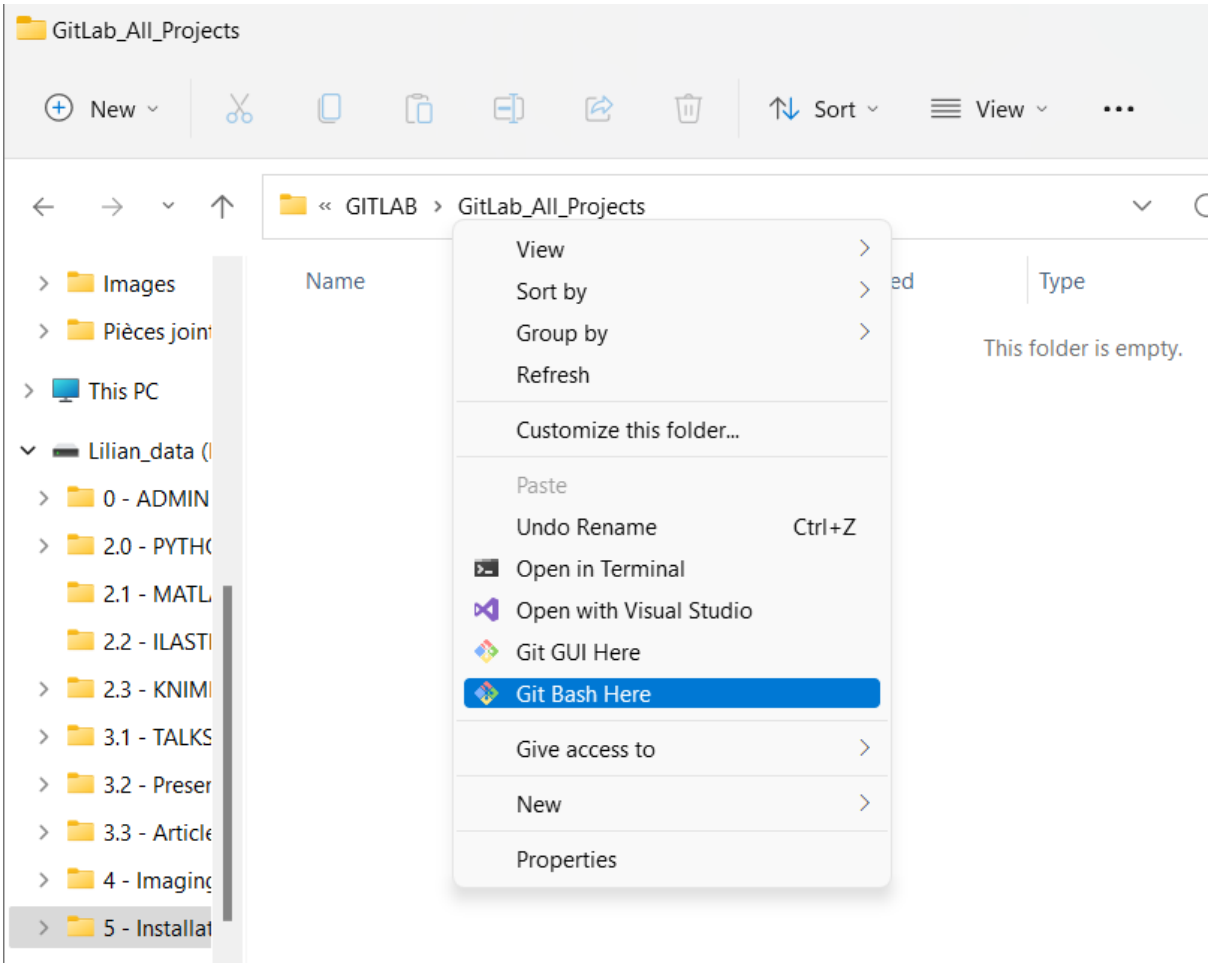
>>> n

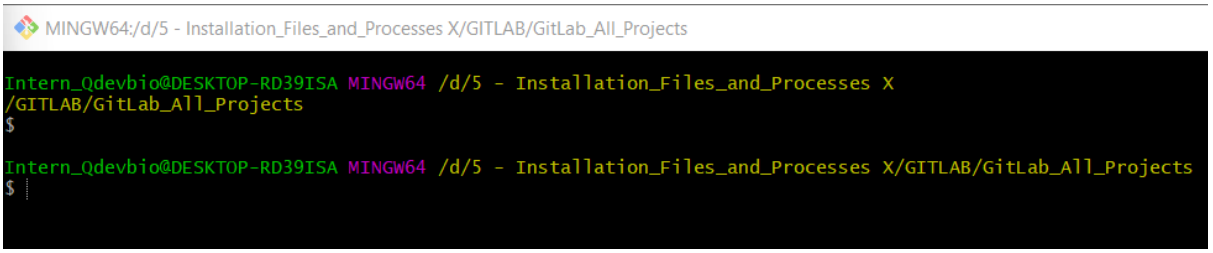


BE CAREFUL : if you overwrite the ssh keys, when you will enter the command in the git bash to clone the project from GitLab to your PC, the permission will be denied so you will have to search for the “id_xxxxx” file you recently overwrote and to add its ssh private key (inside of it) to the GitLab directly (Go to “Preferences” -> “SSH Keys” and “add a new key”).

Step 5 : Go to the target directory where you want to store the project on your own PC

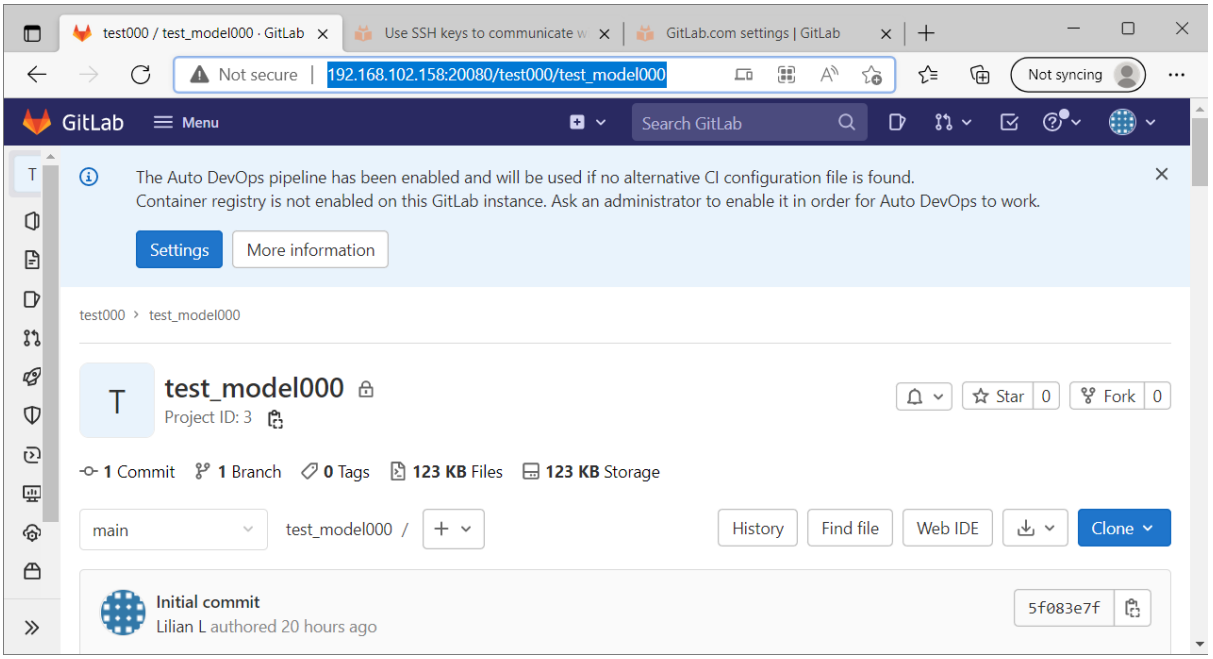
Step 6 : Open a git bash command prompt





Step 7 : Open the gitlab web page of your gitlab project

Step 8 : Copy the address of your project (see in the linkbar)



192.168.102.158

Step 9 : Write the commands below step by step to have the right project address and thus to correctly clone the GitLab project.

The general command is:

>>> git clone ssh://git@192.168.xxx:20022/MyGitLabFolder/MyGitLabProjectFile.git

Tip : press Maj+Inser (or Shift+Insert) on the keyboard to paste something directly in the git bash command prompt.

Step 9.1 : paste this git cloning command :

>>> git clone ssh://git@

Step 9.2 : add the rough project address :

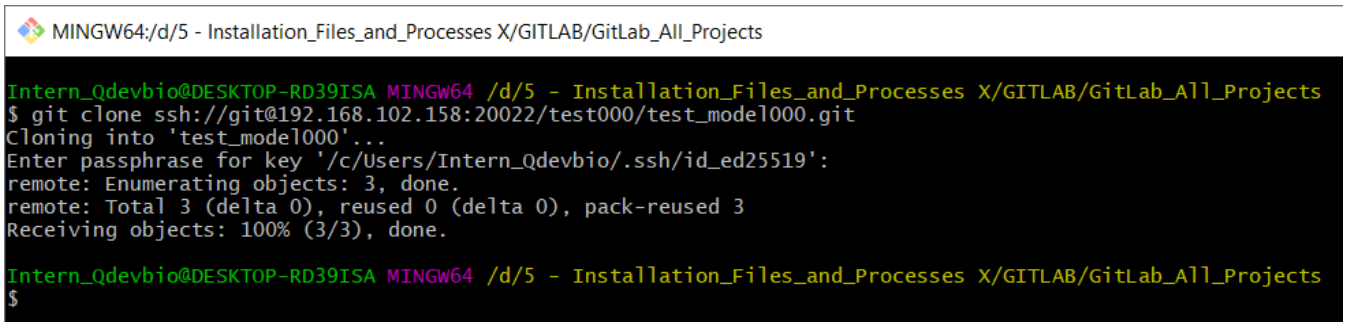
>>> git clone ssh://git@http://192.168.xxx:20080/MyGitLabFolder/MyGitLabProjectFile

Step 9.3 : delete the "http://" base and correct the "80" address part corresponding to the current NAS communicating port with the PC by "22" :

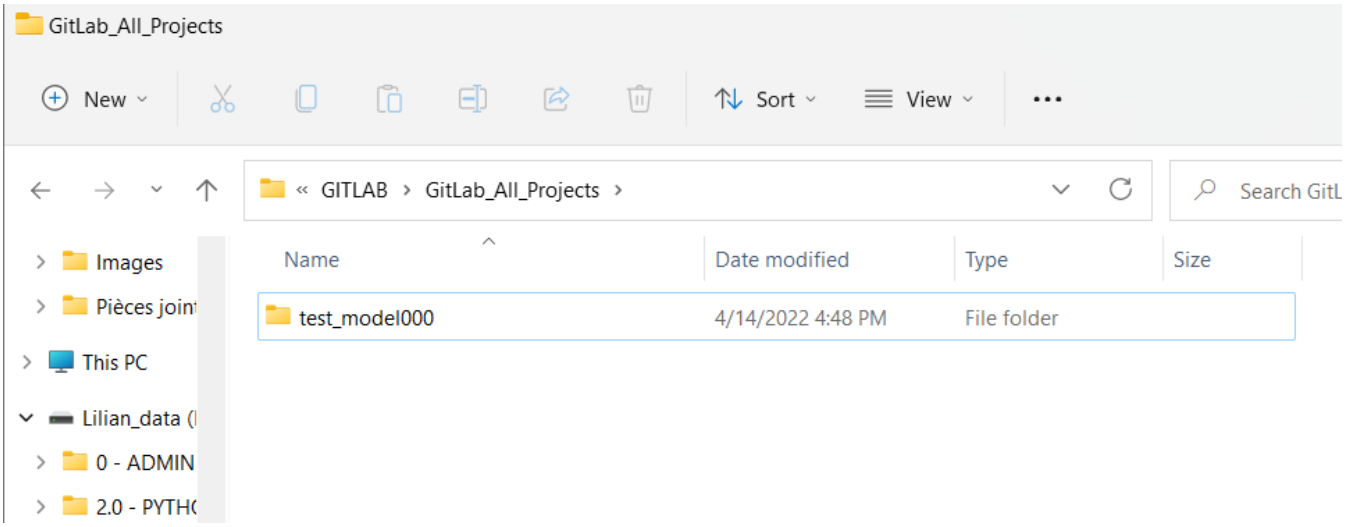
>>> git clone ssh://git@192.168.xxx:20022/MyGitLabFolder/MyGitLabProjectFile

Step 9.4 : add ".git" at the end of the command :

>>> git clone ssh://git@192.168.xxx:20022/MyGitLabFolder/MyGitLabProjectFile.git



Step 9.5 : Enter the passphrase to use the private key and verify that your project has been cloned in your PC target folder :



The Project is cloned.

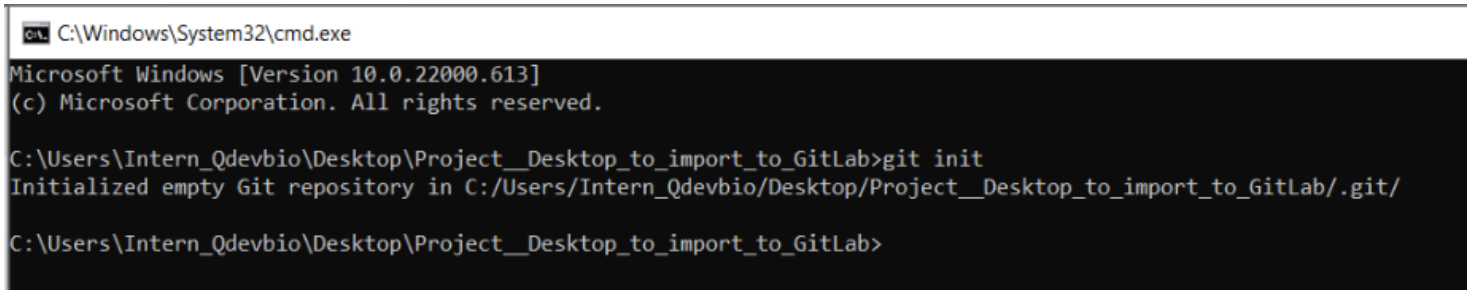
Rough list of all the commands :

git init
ssh-keygen -t ed25519 -C "email@curie.fr"
git clone ssh://git@192.168.xxx:20022/MyGitLabFolder/MyGitLabProjectFile.git

FROM PC TO GITLAB - COMMIT

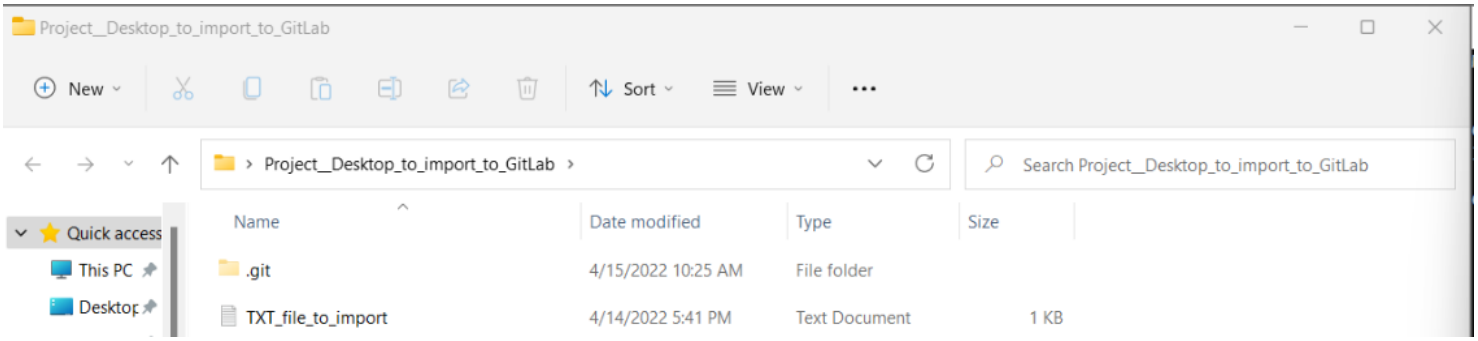
Step 1 : Open the command prompt in the folder containing the files to import.

Type the command “git init” to initialize an empty Git repository.
You can see the hidden files and folders (as the .git is one of them) by clicking on :
“View” => “Show” => ”Hidden items”



>>> git init

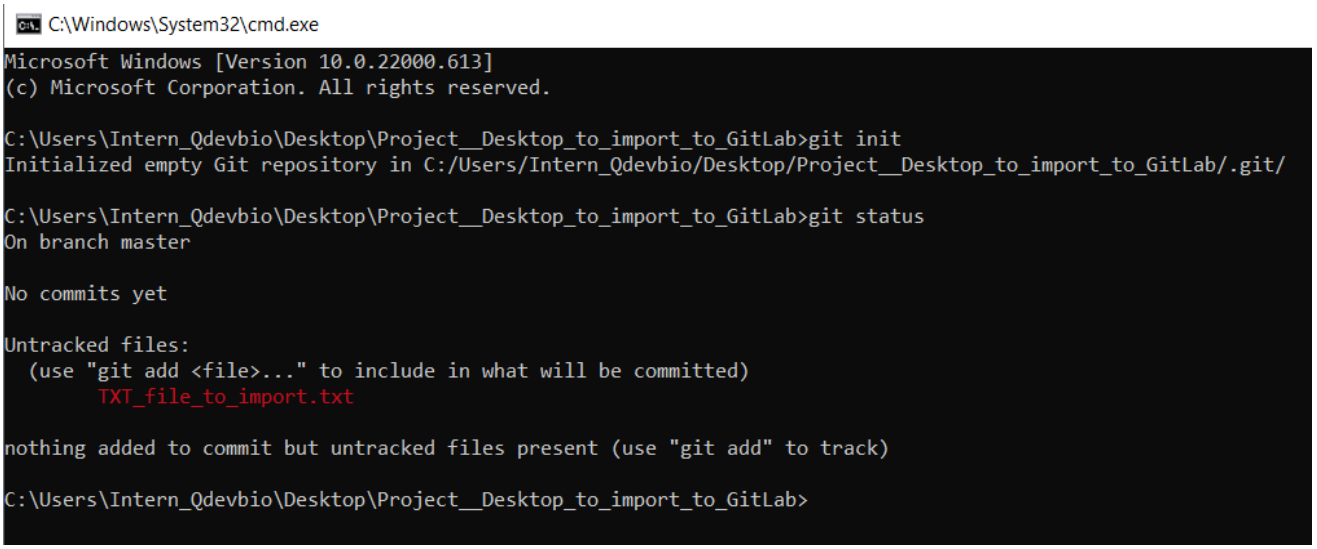
You can see the .git folder created and the file to import below :



Step 2 : Adding the files to upload in GitLab

To see the present files and folders you can add at this moment of the process, type the following command :

>>> git status



Here you see the .txt file that isn't included in the folders and files to commit.
To include a file, type the following command(s) without any quotes :

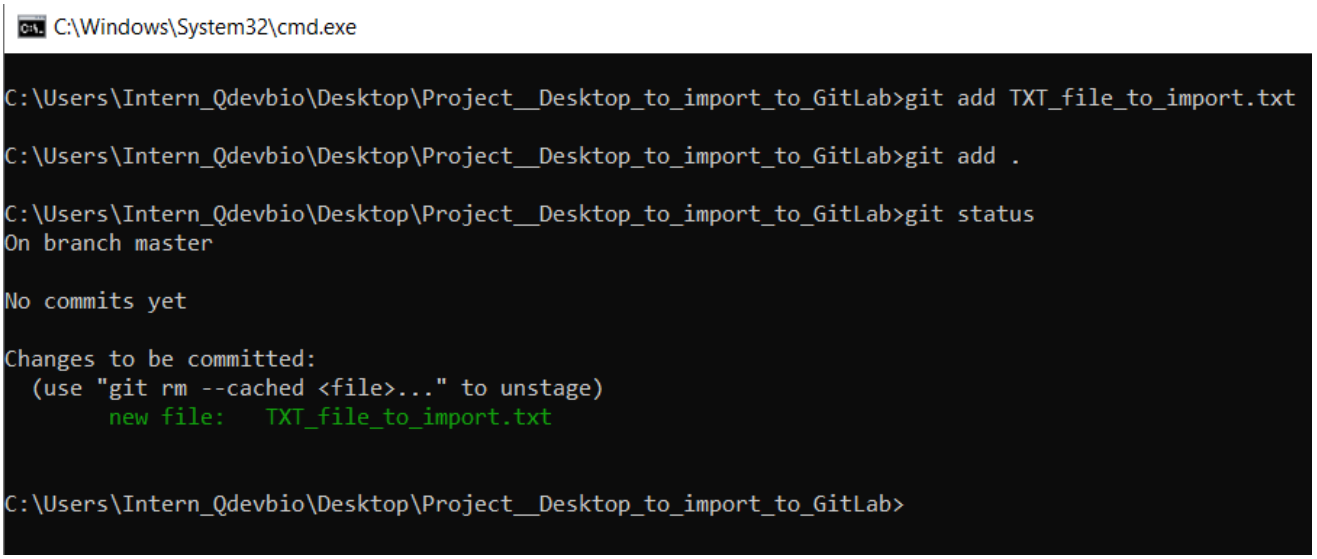
>>> git add TXT_file_to_import.txt

If all the folders and files must be committed, type this command :

>>> git add .

Now you can check for the ready files to be committed by typing this command again :

>>> git status



Step 3 : Uploading the files in GitLab

The following command isn't always necessary but will be needed if it is the first time you upload files and subfolders from the current folder (or if you aren't already registered in the accounts associated with this windows session probably).
You need to indicate the email associated with your GitLab account to correctly authentificate yourself (with quotes) :

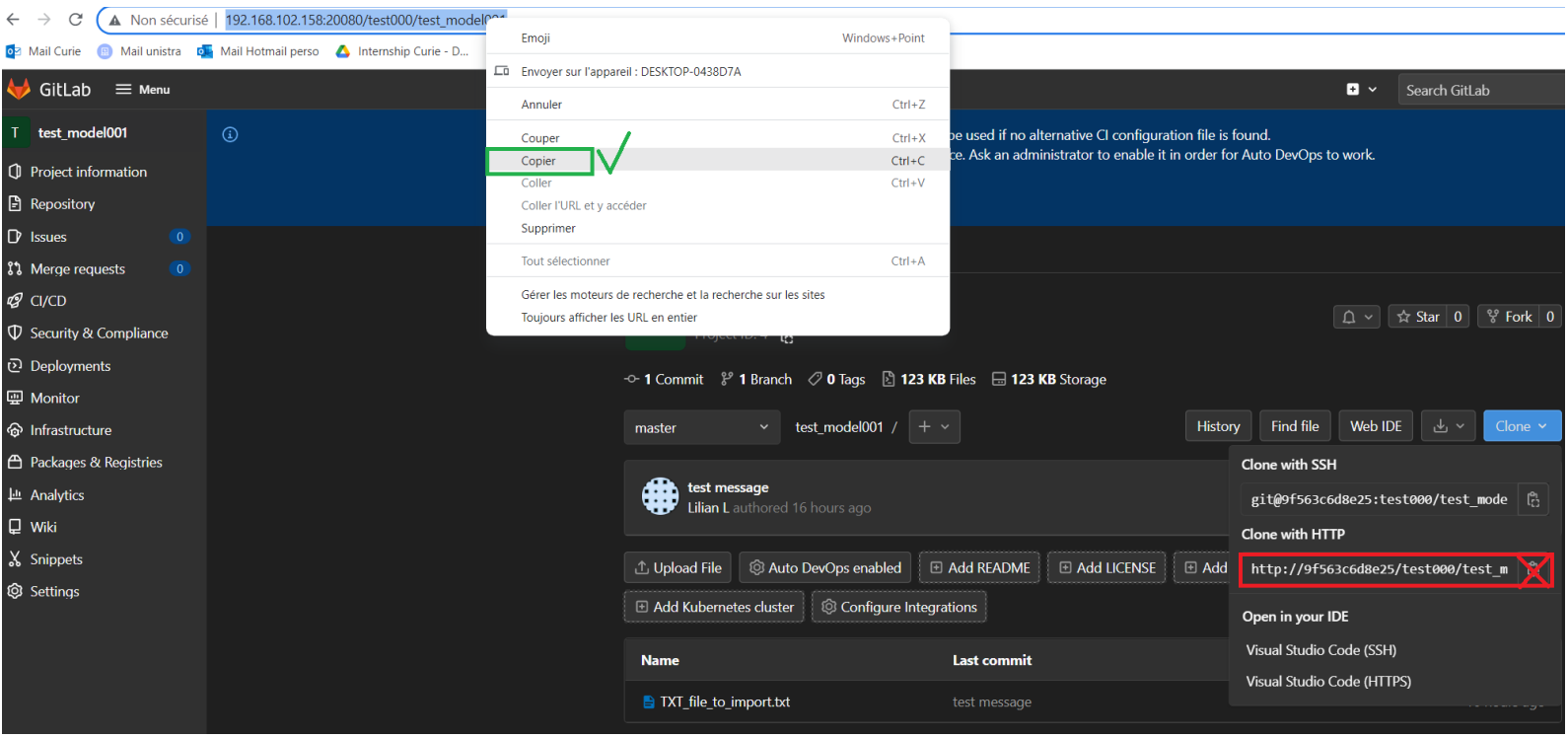
>>> git config --global user.email "email@curie.fr"

Then you can commit the meaningful message associated with your files and subfolders indicating what is the purpose of this global upload (with quotes) :

>>> git commit -m "Message for other users - name of the project"

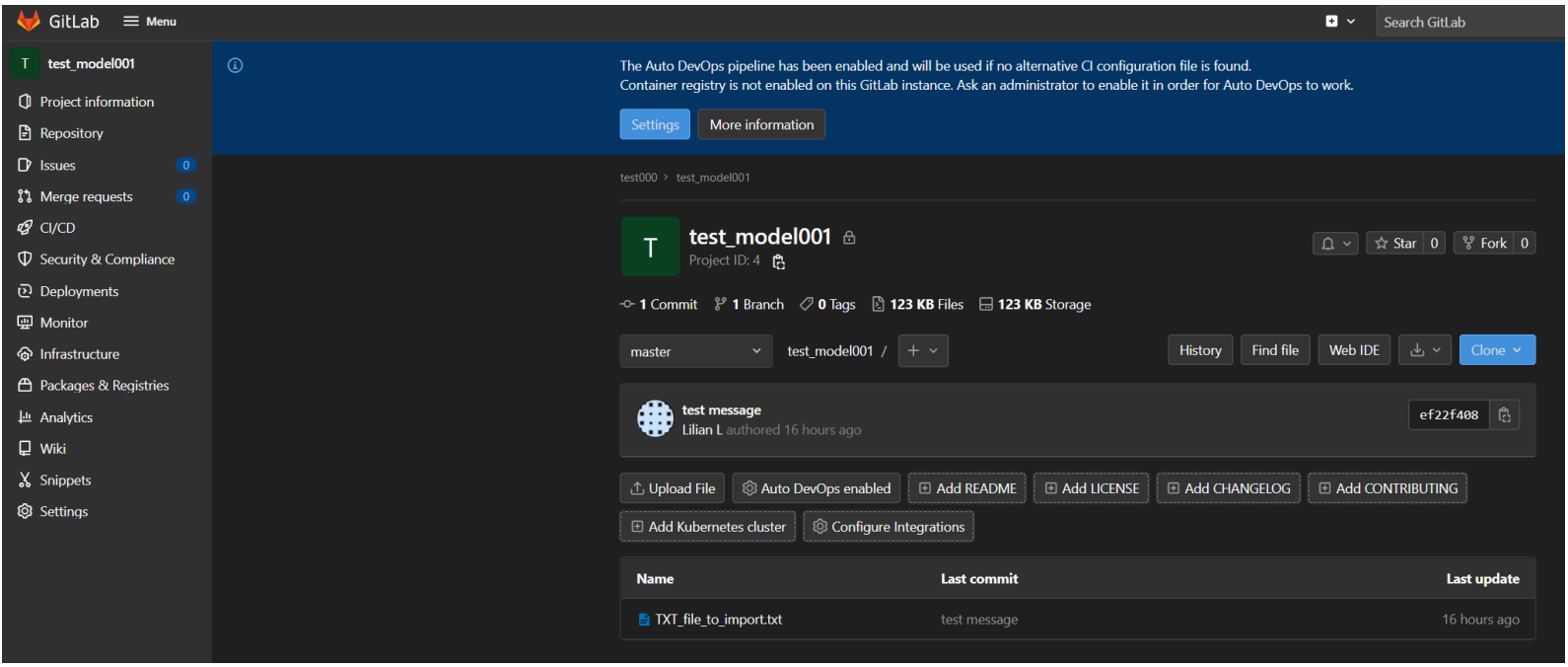
Thus, for the uploading command, you will have to specify the GitLab folder URL where you want to import your files and subfolders, for that you have to copy the URL link simply :

Note : copying the "http" URL when you click on "Clone" in a .git project in GitLab doesn't work for this command.



If there already is an existing project in GitLab (empty or not) at the given URL, it will simply add the files.
If there isn't, it will create a new project and a new .git file named as you named it in the following command :

>>> git push -u "http://192.168.102.158:20080/test000/test_model001.git" master



Rough list of all the commands

git init
git status
git add file.extension
git add .
git commit -m "meaningful message for other users"
git config --global user.email "email@curie.fr"
git push -u "https://xxx.git" master

Useful links

https://docs.gitlab.com/ee/user/ssh.html
https://www.instructables.com/Gitlab-CE-on-QNAP-NAS-Installation-Basic-Usage/
https://github.com/git-for-windows/git/releases/download/v2.35.2.windows.1/Git-2.35.2-64-bit.exe
https://www.openssh.com/portable.html
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
https://docs.oracle.com/en/cloud/cloud-at-customer/occ-get-started/generate-ssh-key-pair.html
Use SSH keys to communicate with GitLab | GitLab
GitLab.com settings | GitLab