Instructions for using Git

February 21, 2019

Contents

1 GIT

This document outlines the basic steps to clone and stay up to date with the CMC-2019 exercises using Git. It is not a comprehensive tutorial on using Git. For detailed tutorials please use the references below.

- Git Cheat sheet
- Interactive tutorial
- Try Git!
- A Visual Git Reference
- git-guide

2 Authorization

In order to pull or push the git repository you need to first set a password to your account. To do this, on your default browser login to the following URL with your epfl credentials.

https://gitlab.epfl.ch/BioRobCMC/2019

Once you open the page, you should see a banner that looks the image below.

Figure 1: HTTPS Git Authorization

Click on the set password and create the password. **It is important that you do this step in order to clone the repository.**

If you do not see this then you have probably already authorized your account.

3 Windows

3.1 Cloning

- If you haven't already then download the Git Bash application from Git

  Note: During installation, choose *Use Git from Git bash only*
- After installation, either from the main menu or on your Desktop you should now find Git Bash application
- Open Git Bash
- Navigate to the location where you want to install the CMC exercises folder. For example, if you want to navigate to your Desktop:

  $ cd `C:/Users/(YOUR_USER_NAME)/Desktop`
- Then execute the following command to clone the repository:

  $ git clone `https://gitlab.epfl.ch/BioRobCMC/2019.git`

This would clone the repository with the default folder name `2019`. If you wish to clone with a different folder name, then use the following command and replace `FOLDER_NAME` with the name of the folder you want to clone into:

$ git clone `https://gitlab.epfl.ch/BioRobCMC/2019.git FOLDER_NAME`

3.2 Pulling

In order to stay up to date with the changes made in the repository, you will have to do a git pull to get the latest version of the exercises.

- Open Git Bash
- Navigate to the exercise directory (Like you did while cloning!) **NOTE :** You need to be anywhere inside the repository directory!!
- Execute the following command:

$ git pull origin master

If you have the latest version then the output of the above command should be:

$ Already up-to-date!

Since this is a public repository for all students, you can not push your changes to the repository.

4 MacOS

4.1 Cloning

- Open Terminal using Spotlight search. You can also go to Applications -> Utilities -> Terminal
- Navigate to the location where you want to install the CMC exercises folder. For example, if you want to navigate to your Desktop:

$ cd `~/Desktop`

- Then execute the following command to clone the repository:

$ git clone `https://gitlab.epfl.ch/BioRobCMC/2019.git`

This would clone repository with the default folder name `2019`. If you wish to clone with a different folder name, then use the following command and replace `FOLDER_NAME` with the name of the folder you want to clone into:

$ git clone `https://gitlab.epfl.ch/BioRobCMC/2019.git FOLDER_NAME`

4.2 Pulling

In order to stay up to date with the changes made in the repository, you will have to do a git pull to get the latest version of the exercises.

- Open Terminal
- Navigate to the exercise directory (Like you did while cloning!) **NOTE :** You need to be inside the exercise directory!!
- Execute the following command:

$ git pull origin master

If you have the latest version then the output of the above command should be:

$ Already up-to-date!

Since this a public repository for all students, you can not push your changes to the repository.

5 Linux

5.1 Cloning

- Open a terminal using your default application search.
- Navigate to the location where you want to install the CMC exercises folder. For example, if you want to navigate to your Desktop:

  $ cd `~/Desktop`

- Then execute the following command to clone the repository:

  $ git clone `https://gitlab.epfl.ch/BioRobCMC/2019.git`

  This would clone a repository with the default folder name `2019`. If you wish to clone with a different folder name, then use the following command and replace `FOLDER_NAME` with the name of the folder you want to clone into:

  $ git clone `https://gitlab.epfl.ch/BioRobCMC/2019.git FOLDER_NAME`

5.2 Pulling

In order to stay up to date with the changes made in the repository, you will have to do a git pull to get the latest version of the exercises.

- Open Terminal
- Navigate to the exercise directory (Like you did while cloning!) **NOTE :** You need to be inside the exercise directory!!
- Execute the following command:

  $ git pull origin master

If you have the latest version then the output of the above command should be:

$ Already up-to-date!

Since this a public repository for all students, you can not push your changes to the repository.

6 Personal Repository

This section describes how you can create your own repository to maintain your code and make the best use of GIT throughout the course. This is not necessary to complete the exercises during the course.

6.1 Creating a new repository

Figure 2 shows the different options while creating a new repository. It is important to set the visibility/accessibility of your repository to Internal as it makes sure that your work is not visible to everyone unless you give someone explicit permissions later on.

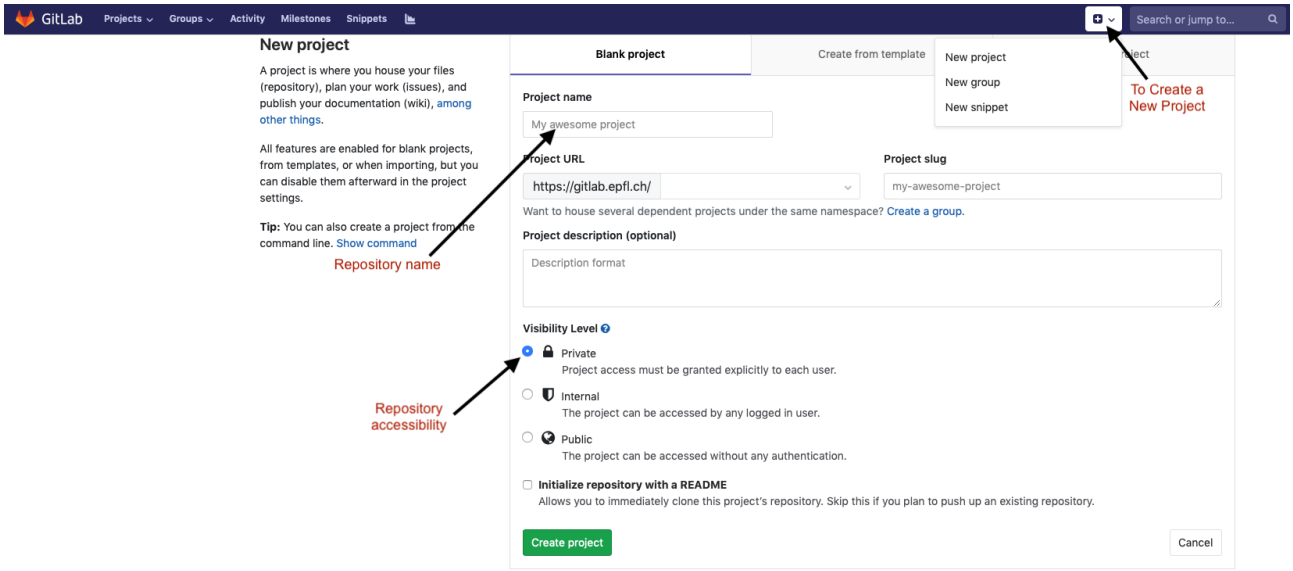

Figure 2: Creating a new git repository

6.2 Cloning

The newly created repository can be cloned to your computer using the same steps described earlier to clone the main exercise repository.

```
$ git clone {REPOSITORY_CLONE_URL}
```

6.3 Status

One of the most important elements to keep track of your cloned repository is to keep track of its status. You can do so at any time by navigating in to the cloned repository on your terminal and then executing the following command :

```
$ git status
```

The output of the command will be explained in the following several sub-sections.

6.4 Pushing

Once you have cloned the repository, you can now start populating your cloned folder with the relevant files. GIT offers several stages in maintaining your files:

6.4.1 Stage 1 - Untracked files

When new files is added for the first time to the cloned repository on your computer, GIT recognizes the new files and add it under the category of untracked files. Meaning GIT will not keep track of any changes made to these files even though they are inside the repository.

6.4.2 Stage 2 - Tracked files

One you decide a particular file needs to be tracked, you need to tell GIT explicitly to do so. The command to do so is the following,

```
$ git add {FILE}
```

The above command creates a snapshot of the file you are interested in. This does NOT mean any changes you make after are kept track of, when ever you think it is important to take a snapshot of the change you made you need to execute this command on every file you are interested in. This basically overwrites the previous snapshot you made unless you committed the files.

6.4.3 Stage 3 - Commit

Once you decided that a particular snapshot that you added (one/several files) need to remembered as part of your history of changes, you need to commit them. You can commit your changes using the command,

```
$ git commit
```

This command will open up your default text editor from the terminal. Here you are expected to write a short message describing the changes you made to the files that you want keep in history. This helps you later on to quickly look at your history messages in a readable format to know the overview of changes made during different stages of development. After you are done, the a snapshot of this history is now saved on your computer locally.

6.4.4 Stage 4 - Pushing

Finally when you decide that the changes you made along with your history should be seen by other members or needs to be stored on the cloud, you need to push the history to the online repository using the command:

```
$ git push
```

The first time you do this you have to tell GIT where you are trying to push the changes using the command,

```
$ git push --set-upstream origin master
```

Where, origin represents that you are trying to push to the default online repository. master represents the main branch of the repository that you are trying to push to.