Setting up your Git and GitLab environment

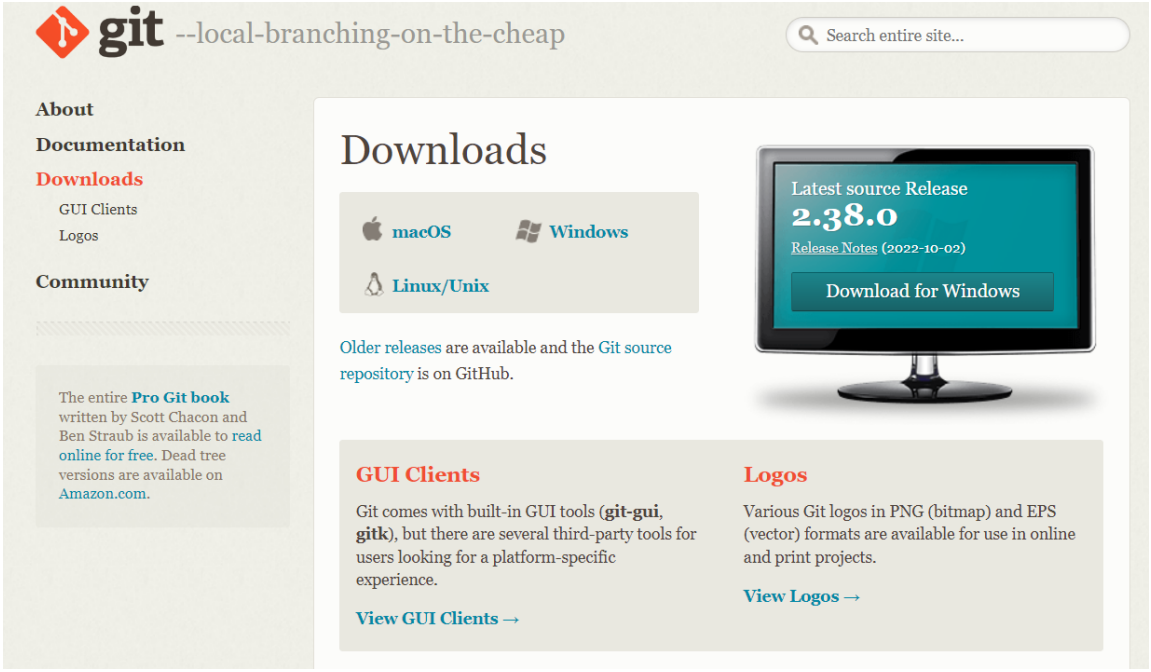
Now is the time for you to harness the power of Git! Let’s start by installing Git on your computer, then you will create and take a grasp of your GitLab account.

Download and install Git

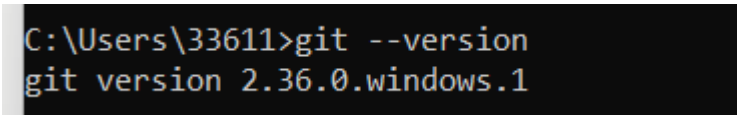
Installing Git on your computer will make it possible for you to clone code repositories from the net, create your own, and access and perform all the actions that a developer needs to interact with a Git-versioned code base.

There are two ways to use Git, the GUI (Graphical User Interface) and the CLI (Command Line Interface). Even though the CLI might seem a bit more daunting and tedious to use, we recommend learning how to use it and getting familiar with it first. Git can be a bit tricky and the only way to grasp it is to make sure what happens under the hood. For that reason, we will teach you how to use the command line in this course, but feel free to switch to the GUI later if you feel it is more suited to your needs.

▼ Download Git from its official website, then install it



▼ Once installed, open a terminal and type ***git --version*** to check that it has been successfully installed



As you know by now, one of the main advantages of Git is its ability to track changes as well as their authors. In order for other collaborators to identify you, you should configure your Git by providing both your full name and your e-mail

```
git config --global user.name "first_name last_name"
git config --global user.email "mail@example.com"
```

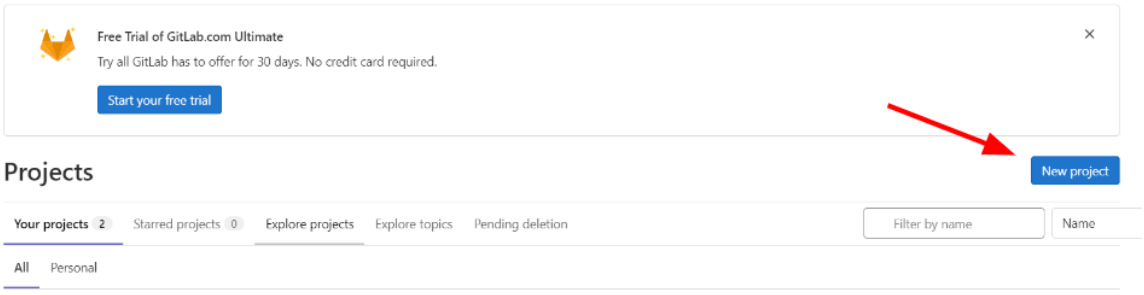
Sign up/in to GitLab and create a blank project

Throughout the course, we will be using GitLab as a remotely hosted platform to push our work and collaborate using Git.

- If you do not already have one, start by creating a GitLab account: https://gitlab.com/users/sign_in

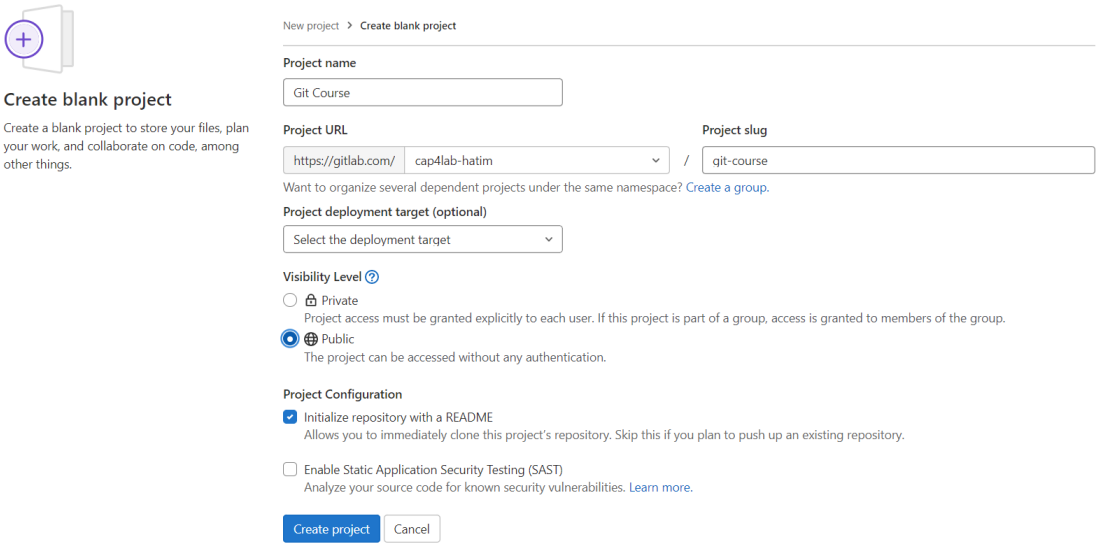
▼ GitLab might guide you through the creation of your very first project. Go to your **dashboard** by clicking on the top left GitLab logo:

▼ Then click on “New Project”:



- Select the first option “Create blank project”

▼ Type “Git Course” as a project name and select “Public” as a visibility level. This way, anyone can view your work. Your project settings should look like this:



- Click on “Create project”. Notice that a default *README.md* file has been added to your project.

Connecting your local Git to your remote repository

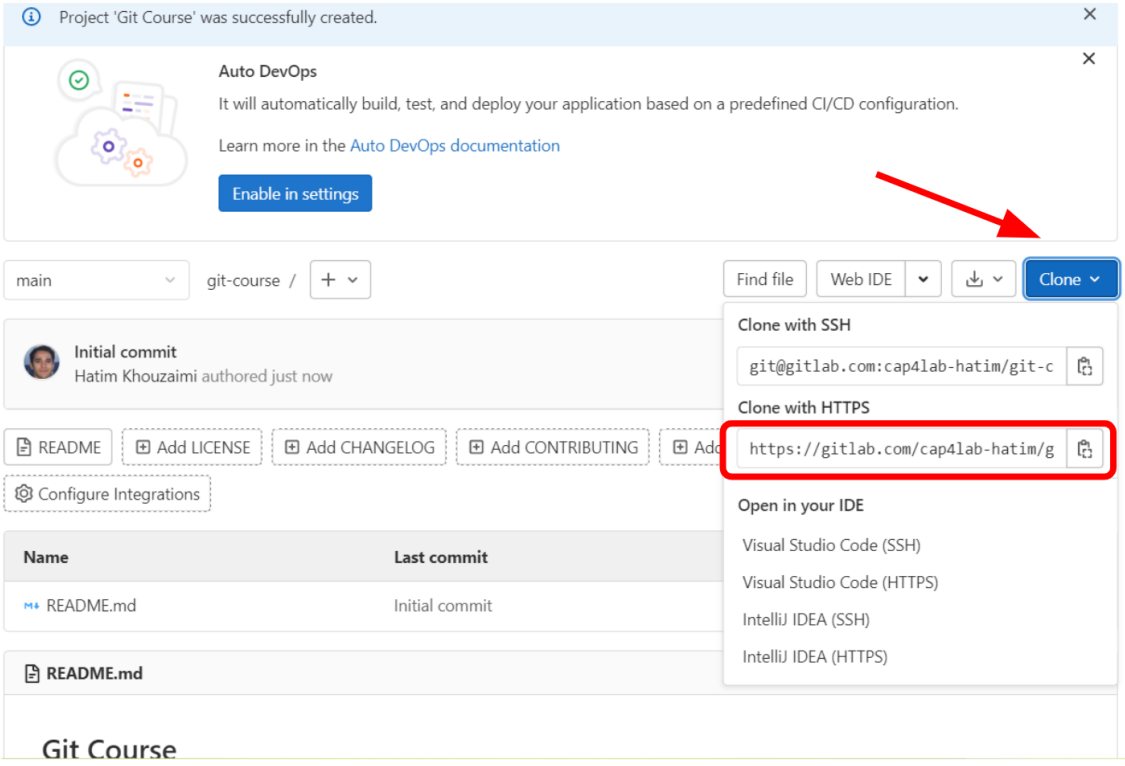
You might find yourself in one of the two following situations:

1. You still have no files in your project, in which case you should **clone** the empty repository you have just created and gradually populate it as you work.
2. You already have already started working on your project and you want to start versioning it using git. In this case, you can follow the steps provided in the default *README.md* file. If your project folder is still not in the form of a Git local repository, use your terminal to navigate to it then use the **git init** command.

Here, we will use the first option since we still have no project. In order to clone the blank repository you have just created:

- Create or choose a folder that will contain the subfolder corresponding to your project (for example: “My Projects”).

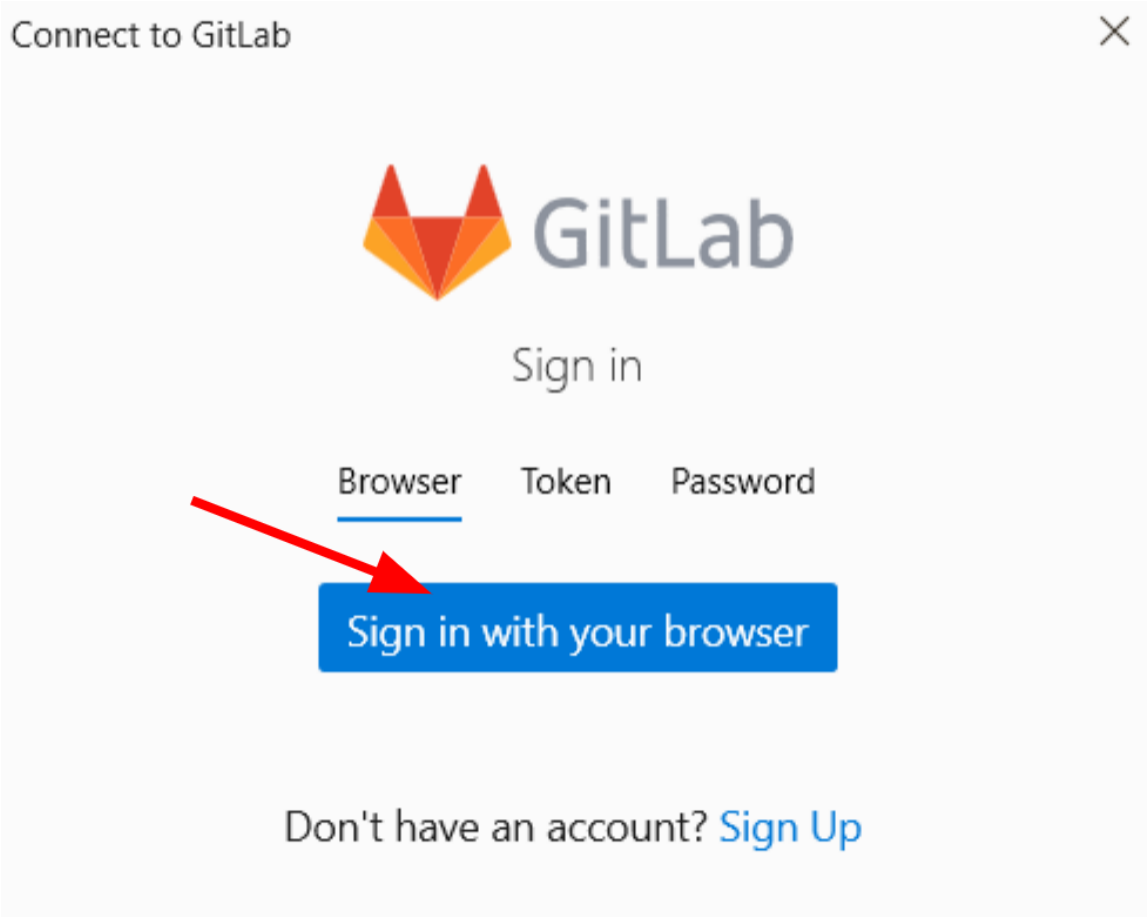
▼ Retrieve the HTTPS address of your repository from GitLab by clicking on the “Clone” button, as follows:



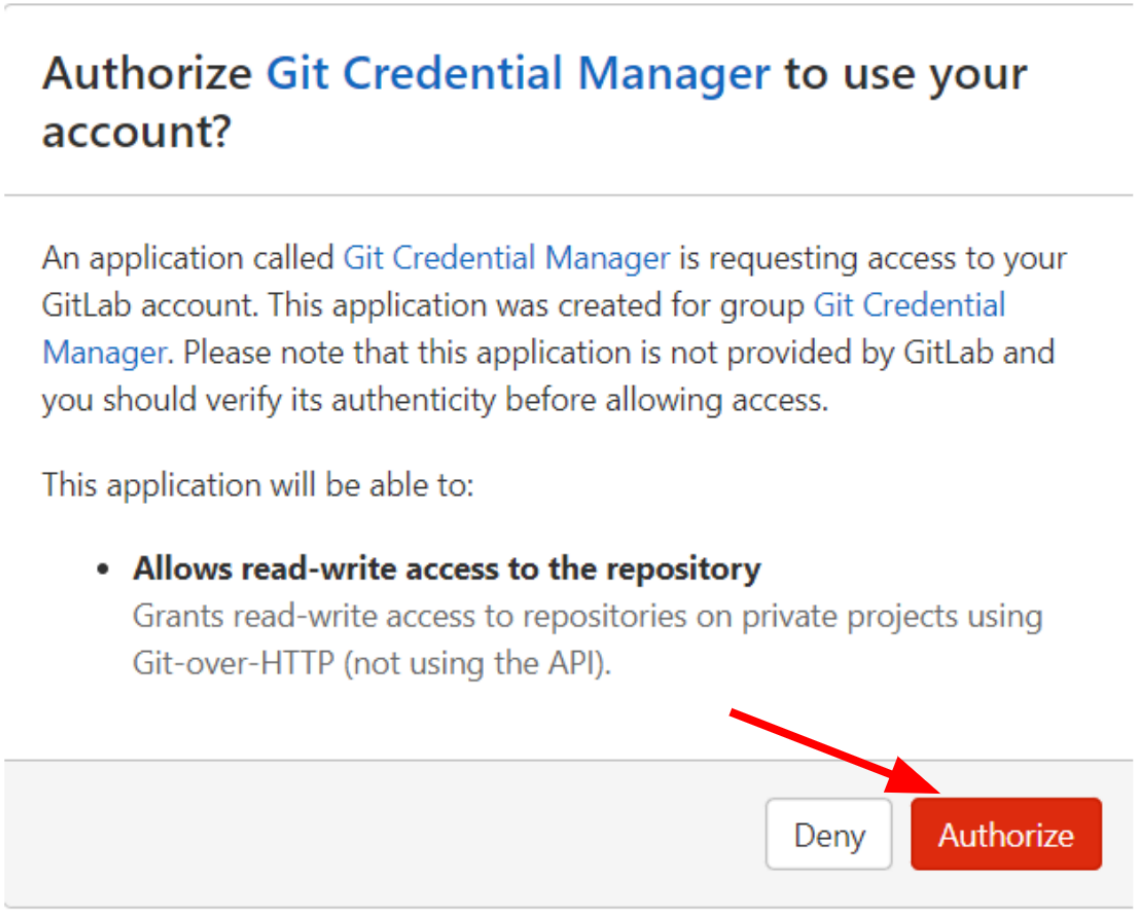
▼ Then use your terminal to navigate to the folder you just created or selected and type the following command (specify your own repository HTTPS address):

```
git clone your_repo_https_address
```

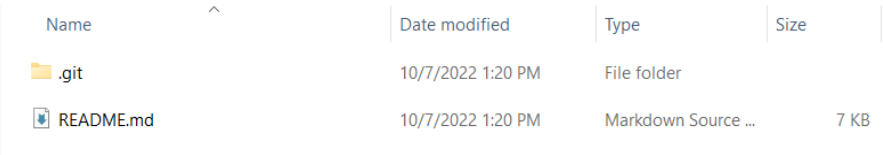
▼ A window will pop up to ask you to sign in with your browser. Click on the corresponding button:



▼ Authorize your Git client to access your repository inside the page you have been redirected to:



▼ Open your project folder and verify that your project has been successfully cloned:



Congratulations! You have just set up both your local and remote environment on GitLab to start harnessing the power of Git. For now, your repository only contains the default *README.md* file. In the next section, we will start populating our project with some files and modifications.

If your file navigator is configured to display hidden files and folders, you might have noticed the presence of a *.git* hidden folder. This folder contains all the information about your repo and can be used to generate any version of your files.