

Git / Github setup for MATH 342W

In the software engineering and data science worlds, git is the standard source control program thus it's worth it to learn now. Further, you'll be submitting your labs and theory homeworks with git and github.

The first step is to make sure you have git on your computer. It comes preinstalled on most flavors on Linux. For the MAC, it is usually preinstalled. Try by running

```
git --version
```

in the terminal. If you see "git version <number>", then it's installed. If you do not see this, you can install git by opening the terminal and running

```
brew install git
```

If you don't have brew installed yet, you first install brew by following the instructions at <https://brew.sh>.

Windows does not have git preinstalled. You should download git for windows at <https://git-scm.com/download/win>. The recommended installer is the "standalone installer" and "64-bit Git for Windows Setup". Once that's installed, you should open up the "Git Bash" program which will provide you a terminal (the regular CMD terminal does not work with git)! On the MAC, you can just use the standard terminal.

In the terminal, navigate to the folder you want to store your class materials. For me this is the "~/workspace" so I run

```
cd ~/workspace
```

We now setup your own security keys. This can be done by typing in the following command

```
ssh-keygen -t ed25519 -C "<your email address>"
```

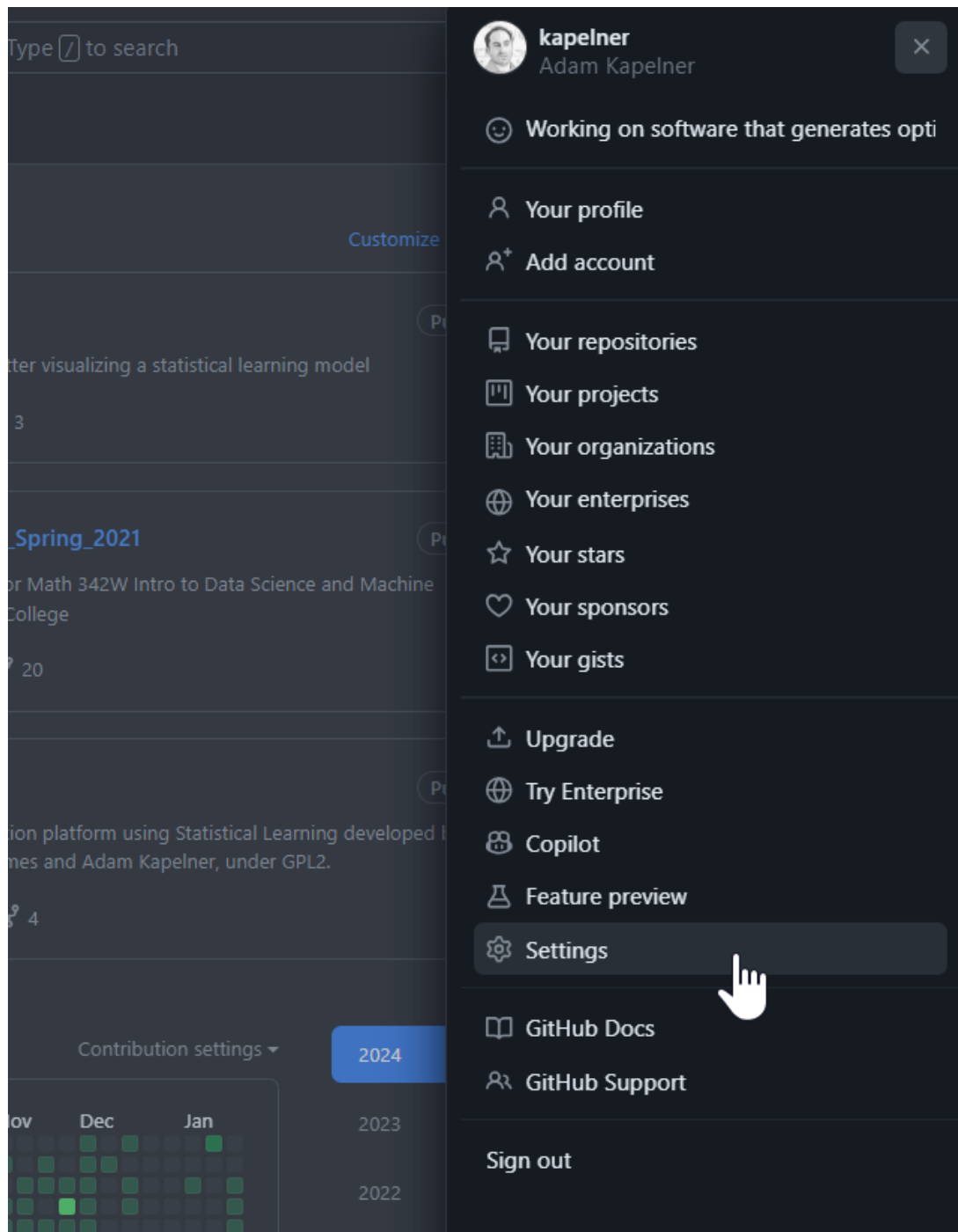
I use no password so I just press enter twice. This is more convenient, but less secure as if someone gains access to your laptop, they can impersonate your github account and whatever else uses these keys. If the key creation is successful, you should see the following message (thanks to Hadassah):

```
hadassah@krigsman:~/Documents/GitHub$ ssh-keygen -t ed25519 -C "krigsman@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/hadassah/.ssh/id_ed25519):
Created directory '/home/hadassah/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadassah/.ssh/id_ed25519
Your public key has been saved in /home/hadassah/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:nfFYpsHLjk5roP1kuut5BRsVcvWtUsTj4edUvYUejSg krigsman@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|      . oo.o.+ |
|      +E .oB.= |
|      .+.o+o+= |
|      oo @ .+oo |
|      S+B o .+ |
|      . .o. . . |
|      o .=. . |
|      .Bo. |
|      .***+ |
+----[SHA256]-----+
```

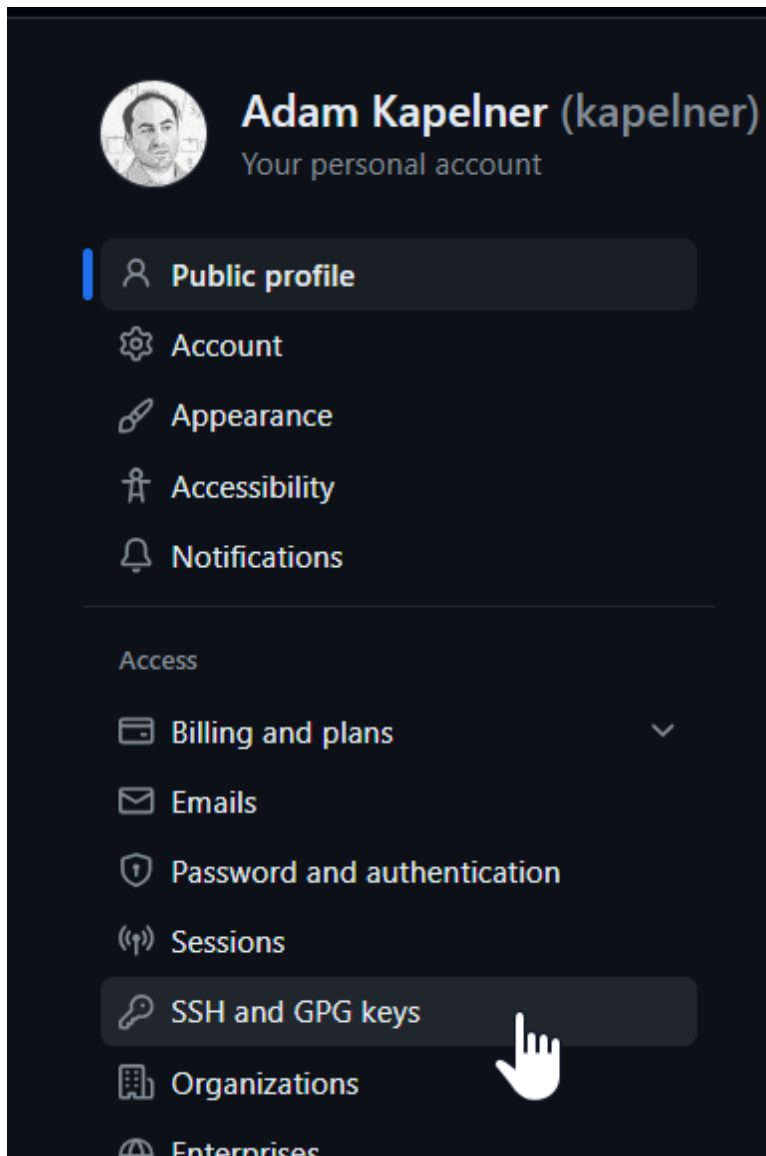
If you don't already have a github.com account, you create one. Use a username that's appropriate! Employers will see it! Then login to github.

Now, send me a direct message on slack with your github homepage link. For me that's <https://github.com/kapelner>. I need this to be able to grade your assignments!

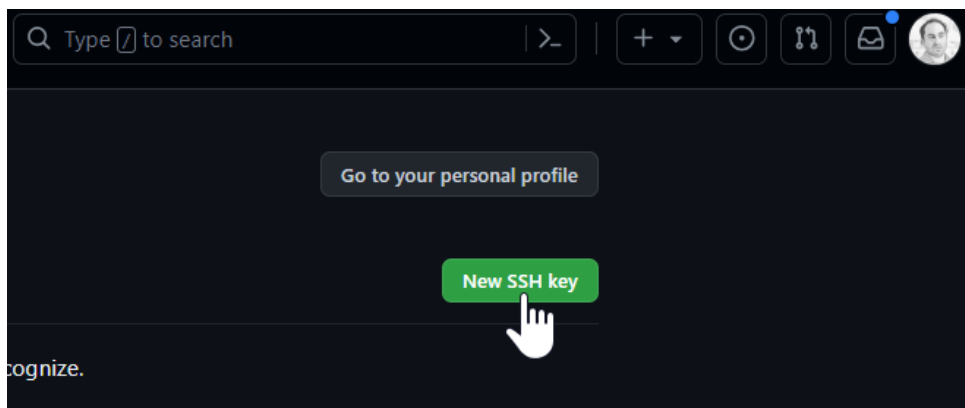
We then add your personal security key to github. You can do this by clicking on your account icon in the top right corner of your account homepage and clicking settings:



Afterwards you'll click on the "SSH and GPG keys" on the left menu:



Now click the “New SSH key” in the top right



The title is up to you. The key belongs to a computer, so if you have multiple computers, you may want to name this key with something that identifies the computer. I use “my_lab_laptop”.

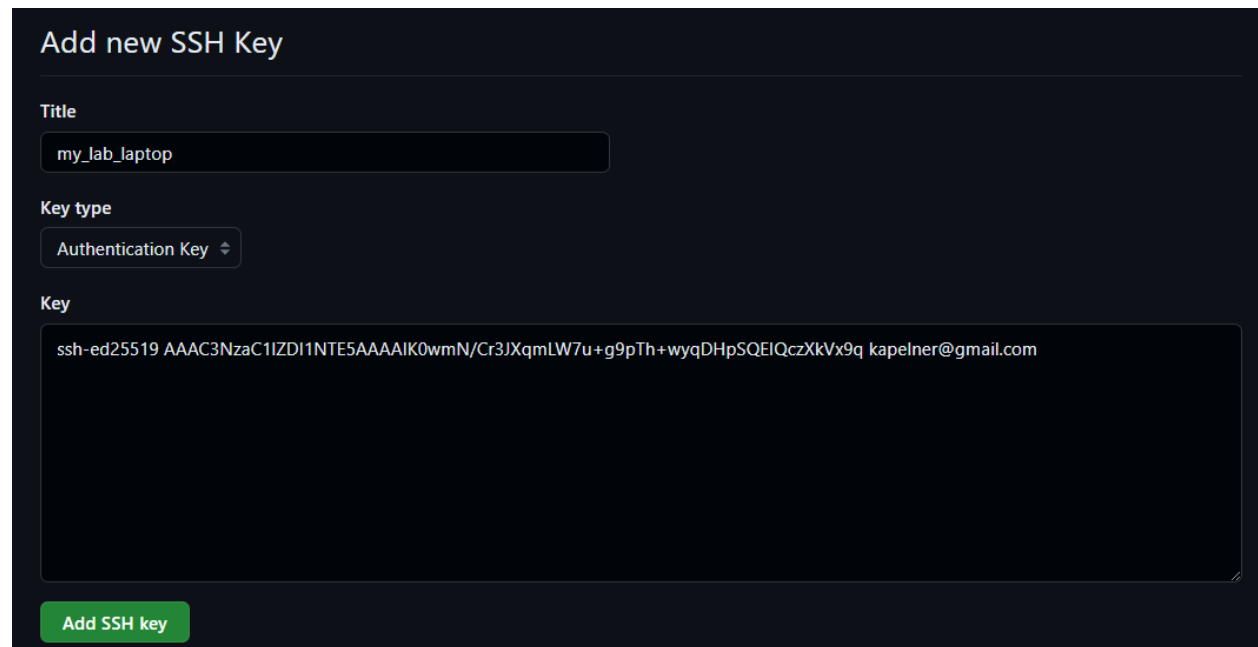
Now you need to open your public key that you generated. I use the following

```
vi ~/.ssh/id_ed25519.pub
```

You should see something like this

```
ssh-ed25519
AAAC3NzaC1lZDI1NTE5AAAAIK0wmN/Cr3JXqmLW7u+g9pTh+wyqDHpSQEIQczXkVx9q
kapelner@gmail.com
```

Now you need to copy that text exactly and paste it into the “key” field so the final screen should look like this:



Add new SSH Key

Title
my_lab_laptop

Key type
Authentication Key

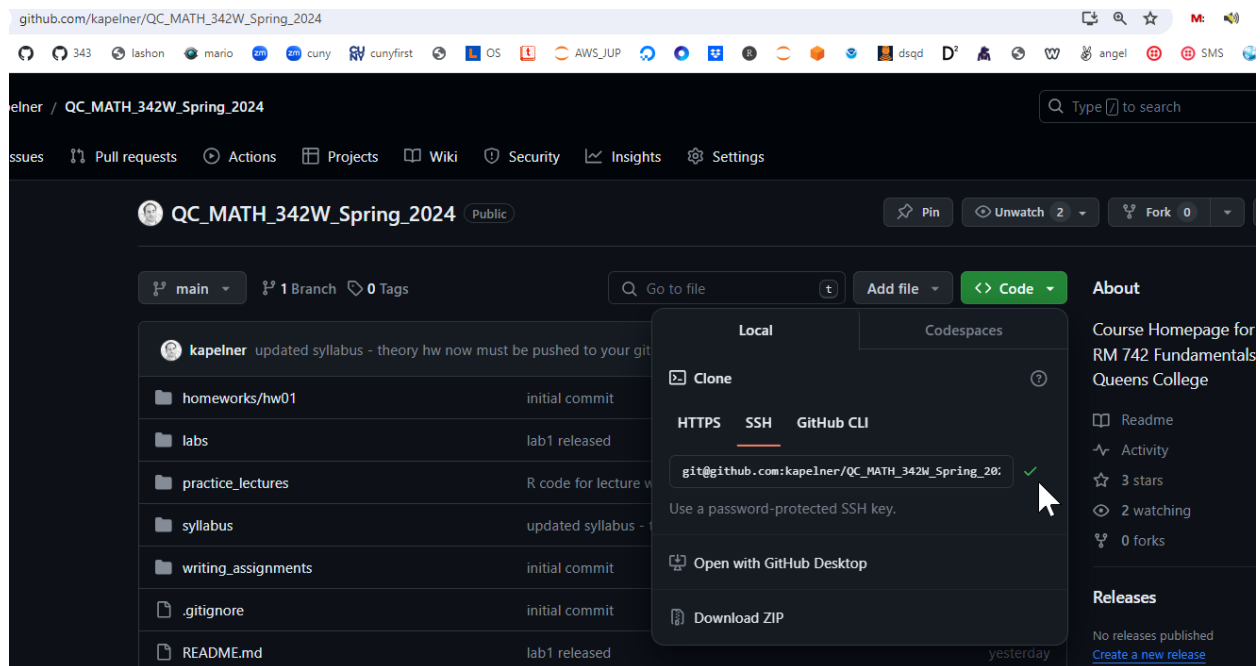
Key
ssh-ed25519 AAAC3NzaC1lZDI1NTE5AAAAIK0wmN/Cr3JXqmLW7u+g9pTh+wyqDHpSQEIQczXkVx9q kapelner@gmail.com

Add SSH key

Now press “Add SSH key”.

Github is now setup with your computer. If you have multiple computers, you need to repeat the two steps (a) key generation, (b) add key to github.

Now let’s get the MATH 342W class materials from the course homepage (which itself is a git repository). You can navigate to the course homepage with your browser. Then click the green “Code” button and select the middle “SSH” option and click the copy icon on its right. This should then look like the following with a checkmark:



Now go to your terminal which is in the folder of your class materials and run

```
git clone [paste]
```

where if pastes successfully should yield the following command

```
git@github.com:kapelner/QC_MATH_342W_Spring_2024.git
```

Press enter to run. If it complains “are you sure...” just type “yes”.

The clone function is your first git command you should know. It essentially downloads the repository of code to your computer. You should now have the QC_MATH_342W_Spring_2024 folder. Check to make sure with file explorer or running

```
ls
```

In the future, as I update course materials, you should download the changes I make. You’ll know I made changes because you’ll see a message in slack that I’ve updated the repository. You can download the updates I made navigating to the folder

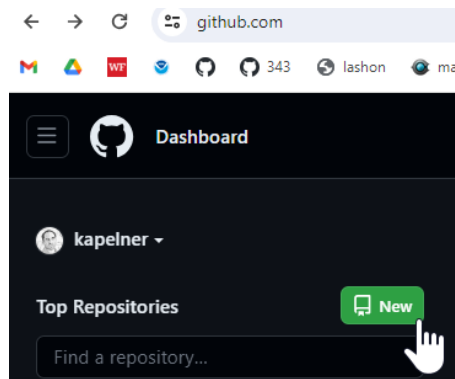
```
cd QC_MATH_342W_Spring_2024
```

and running the pull command:

```
git pull
```

The git pull command is essentially a way to update the repository on your computer from the main repository on github. You will have access to read course files using clone and pull but you don’t have access to change course files using commit (more on that later).

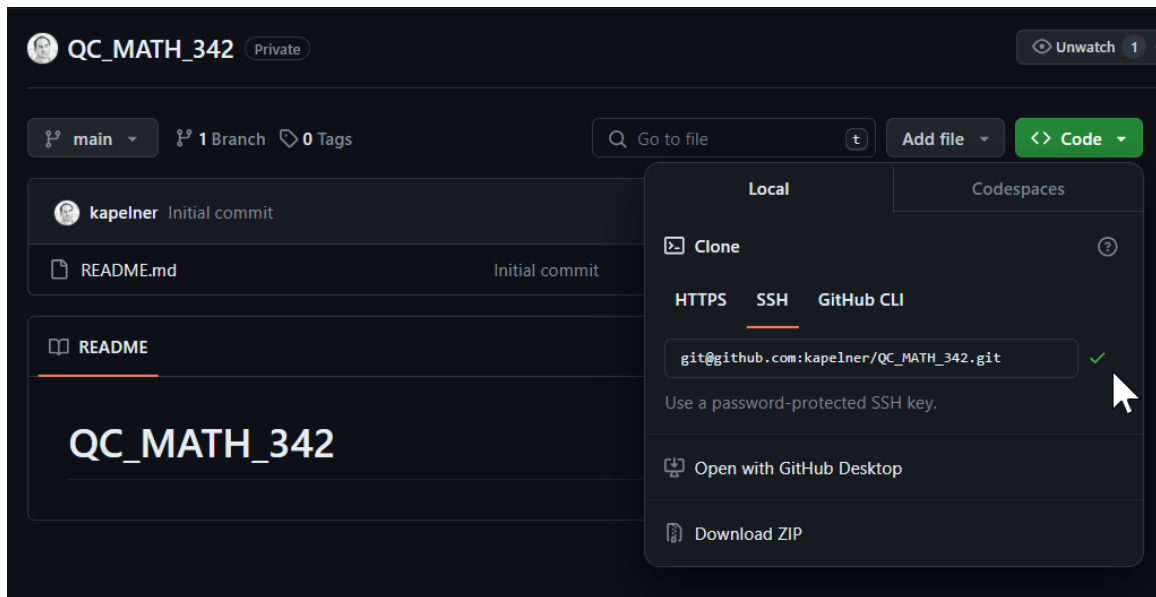
Okay now you are ready to create your own repository, the repository that will store all your homeworks and labs. Go to github.com and click the green new button:



Under “repository name” type “QC_MATH_342”. Then click the “private” button (we will do this for now to prevent cheating). Then check the box “Add a README”. That checkbox is required as otherwise you won’t be able to clone. Your screen should look like:

A screenshot of the 'Create a new repository' page on GitHub. The title 'Create a new repository' is at the top. Below it is a subtitle: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A note states: 'Required fields are marked with an asterisk (*).'. The form has two main sections. The first section is for the repository name, with a dropdown for the 'Owner' (currently 'kapelner') and a text input for the 'Repository name' (currently 'QC_MATH_342'). Below the name input, a green checkmark indicates 'QC_MATH_342 is available.'. The second section is for the repository's visibility, with two radio buttons: 'Public' (selected) and 'Private'. Below these is a section for the repository's description, with a text input field. At the bottom, there is a section titled 'Initialize this repository with:' with a checked checkbox for 'Add a README file' and a link to 'Learn more about READMEs.'.

On the bottom, click the green “Create Repository” button. You should now be taken to the repository homepage. As before, we want to clone it. So click the green “Code” button and then the SSH option and click the copy button until you see the green check mark:



Now let's clone it. Go back to your main directory where you want to store your files for me that's done by typing the following in the terminal:

```
cd ~/workspace
```

Now we clone, type the following

```
git clone [paste]
```

For [paste] you'll press ctrl + v to paste your ssh link that you just copied from your repo page. My link is "git@github.com:kapelner/QC_MATH_342.git" but yours will be different as your username will replace my username, "kapelner". You should see the following:

```
kapel@LAPTOP-J2T9TGGB MINGW64 ~/workspace
$ git clone git@github.com:kapelner/QC_MATH_342.git
Cloning into 'QC_MATH_342'...
Warning: the ECDSA host key for 'github.com' differs from the key for the IP address '140.82.114.4'
Offending key for IP in /c/Users/kapel/.ssh/known_hosts:1
Matching host key in /c/Users/kapel/.ssh/known_hosts:39
Are you sure you want to continue connecting (yes/no)? yes
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Now you will have that folder on your computer. Verify it with file explorer. Now we navigate into the folder

```
cd QC_MATH_342
```

You should now make sure git knows your email address and name by running

```
git config --global user.email "<your email address here>"
```

```
git config --global user.name "<your full name here>"
```

You can now create directories for your assignments:

```
mkdir homeworks
```

```
mkdir labs
```

Now we'll copy the lab assignment from my class repository into your repository

```
cp ../QC_MATH_342W_Spring_2024/labs/lab01.Rmd labs/
```

We will now add this file to your repository by running

```
git add .
```

The "add" command adds new files. You only need to run it when you create new files, not when you edit old files. You can make sure lab01.Rmd is added by running

```
git status
```

You should see it listed in green. The "status" command tells you if anything has changed.

```
kape1@LAPTOP-J2T9TGGB MINGW64 ~/workspace/QC_MATH_342 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   labs/lab01.Rmd
```

Now we will commit this file. Always write comments that make sense.

```
git commit -am "added lab01 assignment"
```

For some reason the quotes don't copy well from Microsoft Word. Another thing Microsoft gets wrong. So type the command yourself manually if it didn't work. You should see the following if it ran successfully:

```
kape1@LAPTOP-J2T9TGGB MINGW64 ~/workspace/QC_MATH_342 (main)
$ git commit -am "added lab01 assignment"
[main 23a6444] added lab01 assignment
1 file changed, 308 insertions(+)
create mode 100644 labs/lab01.Rmd
```

Now we're ready to push these commits to github via

```
git push
```

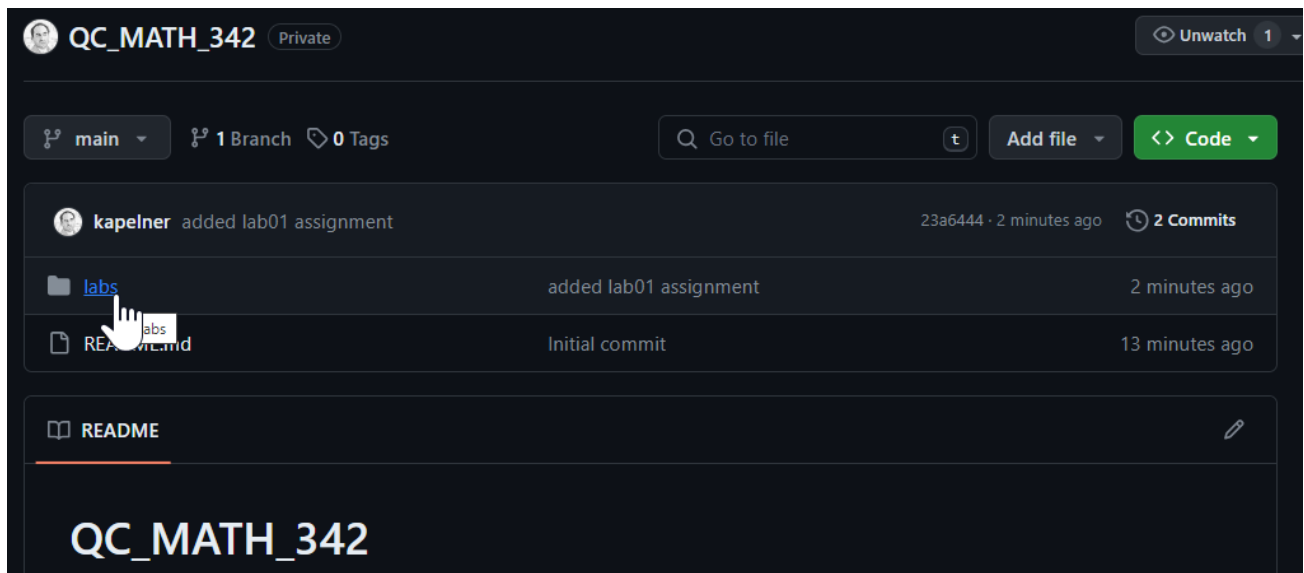
If this worked successfully you should see


```

kapel@LAPTOP-J2T9TGGB MINGW64 ~/workspace/QC_MATH_342 (main)
$ git push
Warning: the ECDSA host key for 'github.com' differs from the key for the IP address '140.82.113.3'
Offending key for IP in /c/Users/kapel/.ssh/known_hosts:3
Matching host key in /c/Users/kapel/.ssh/known_hosts:39
Are you sure you want to continue connecting (yes/no)? yes
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 4.13 KiB | 4.13 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:kapelner/QC_MATH_342.git
   67773e6..23a6444  main -> main

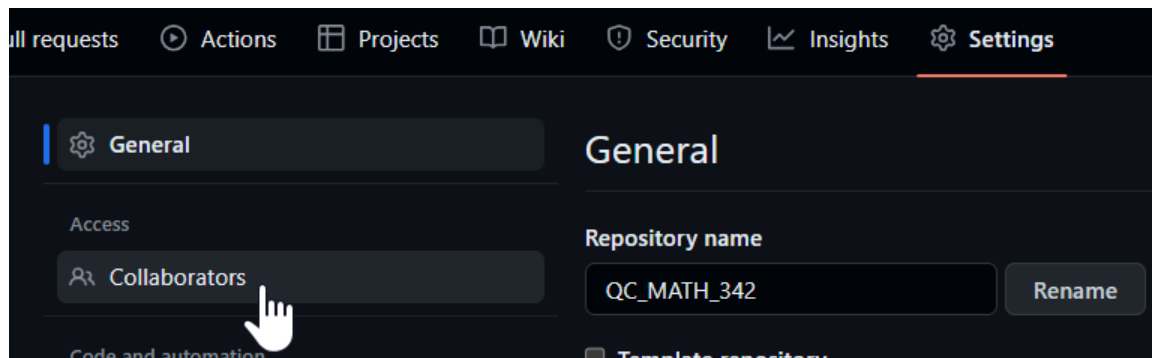
```

Now if you refresh your github page you'll see the new directory:

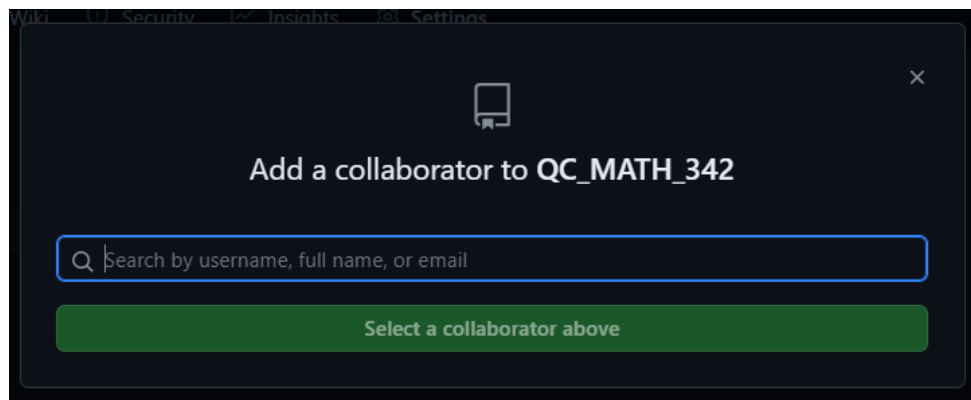


In the future, you'll repeat this git add, git commit, git push sequence for each new assignment. For the theory homeworks, you'll add the PDF of your scanned homework to the homeworks directory and name it "hw0x.pdf". Follow instructions so we can grade your homeworks and labs without searching for them!

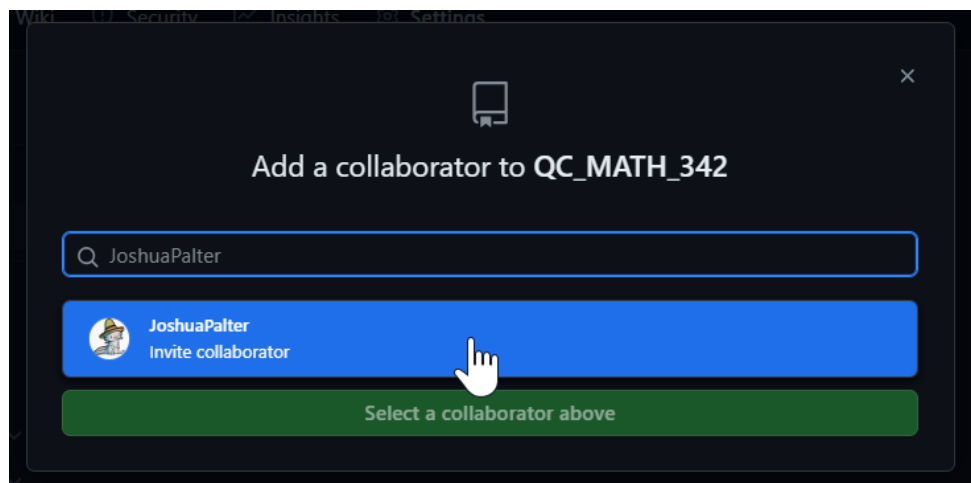
The last step is to add me and the TA as a collaborator on your private repo. This allows us to access your files and grade your assignments. If you don't do this, you will obviously not get credit for any of your homeworks. On the repository page, click "Settings" and then on the left menu, click "Collaborators"



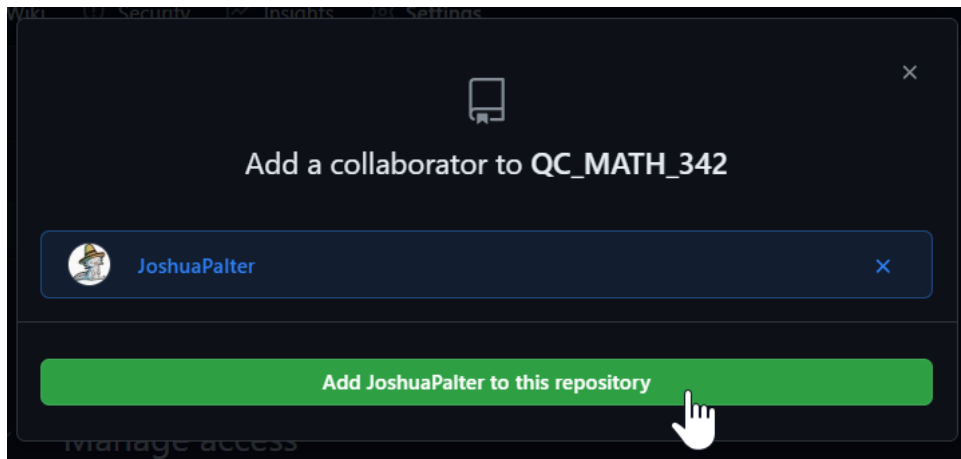
Now it will ask for your password again. Then click the green “Add People” button and you’ll see:



You’ll type in “kapelner” for me and click my account when it comes up. Then click “Add kapelner to this repository”. Then you’ll type “JoshuaPalter” for the TA and click his account when it comes up:



And click “Add JoshuaPalter” to this repository



You should see both of us as “pending Invite”.

Congrats you’re now all set up! There are many great resources about git you can find online if you want to learn more. For this class, you’ll likely only use “clone, add, commit, push, pull, status” but there’s so much more.