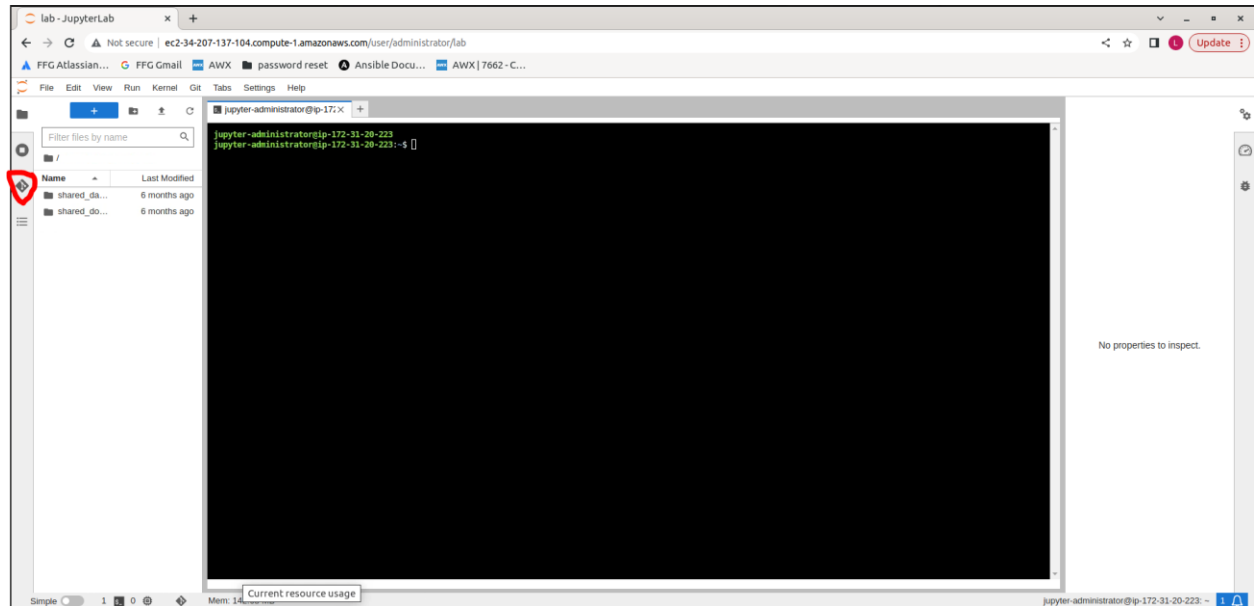


## Using GIT with JupyterHub:

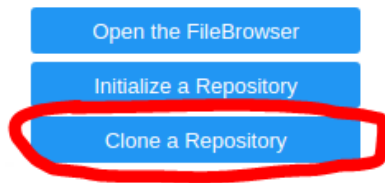
### A. Git Clone:

1. Press the GIT icon in JupyterLabs:



2. Press **Clone a Repository:**

You are not currently in a Git repository.  
To use Git, navigate to a local  
repository, initialize a repository here,  
or clone an existing repository.



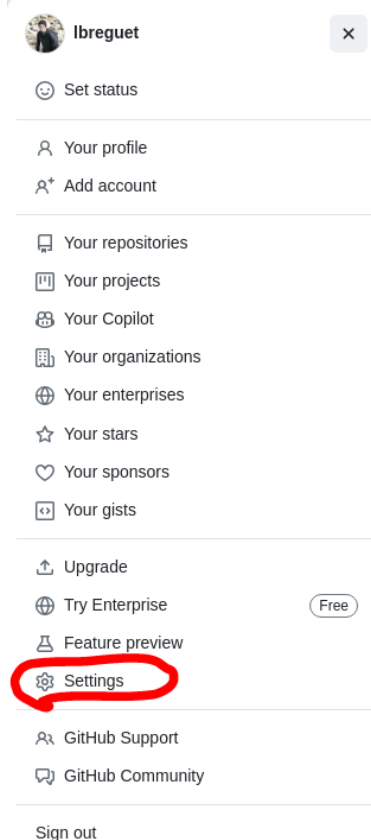
3. Paste the https link to your repo and press **Clone:**

### Clone a repo

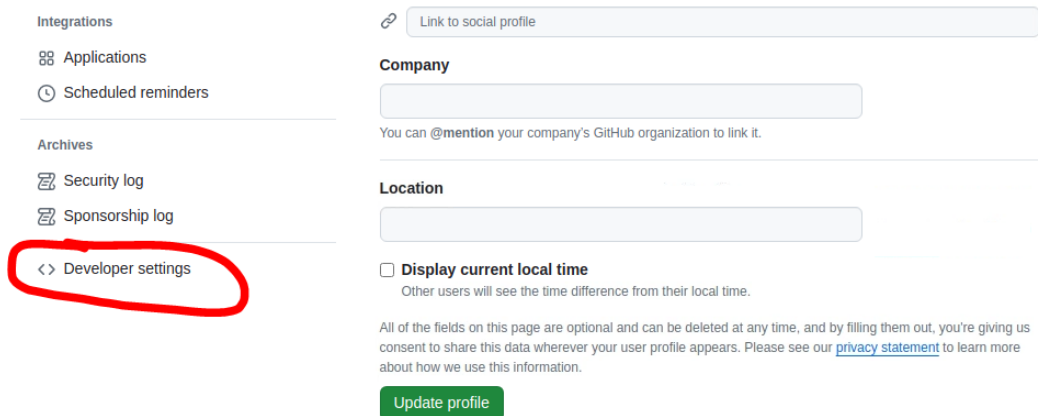
Enter the URI of the remote Git repository

☒ Include submodules  
☐ Download the repository

4. In this next window you need to generate a **Personal Access Token** from Github, so go to Github and go to **Settings** located when clicking your profile photo:



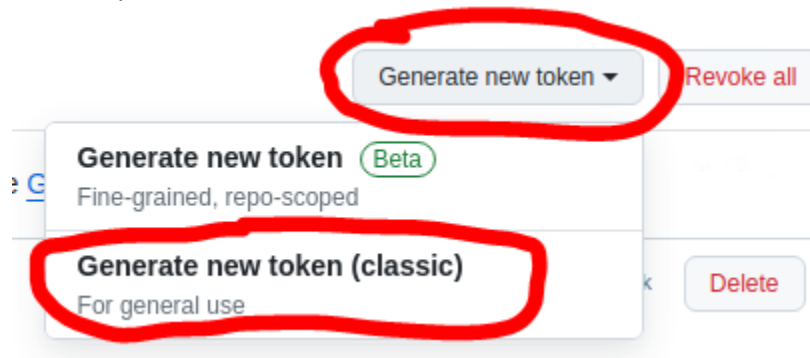
5. Scroll to the last item on the sidebar and click **Developer Settings**:



6. Click on **Personal Access Tokens** and then on **Tokens (Classic)**:



7. At the top, click on **Generate new token** and then click on **Generate new token (classic)**:



8. Give your token an arbitrary name and give it full repo access:

### New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

#### Note

token-jupyter

What's this token for?

#### Expiration \*

7 days

The token will expire on Fri, Apr 26 2024

#### Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

9. Scroll all the way to the bottom and press **Generate Token**

10. Copy the new token (save in a text file):

### Personal access tokens (classic)

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

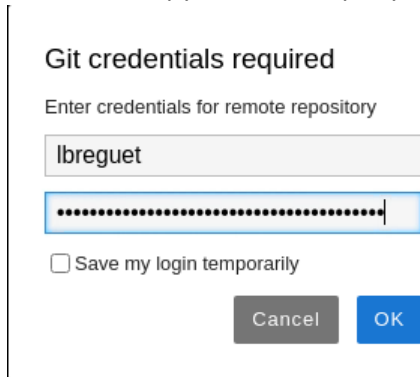
Make sure to copy your personal access token now. You won't be able to see it again!

Copied!

✓ ghp\_vIBAXjrkQiBumqEyKtNjPNf0du9qwj4f82D5 ✓

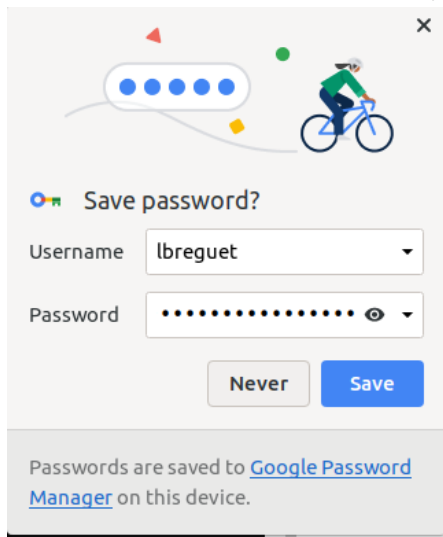
Delete

11. Go back to JupyterLab and input your Github username, paste your token, and press **OK**:



A dialog box titled "Git credentials required" with the subtitle "Enter credentials for remote repository". It contains two input fields: the first is labeled "Username" and contains the text "lbreguet"; the second is labeled "Password" and contains a series of dots. Below the password field is a checkbox labeled "Save my login temporarily" which is unchecked. At the bottom are two buttons: "Cancel" and "OK".

12. Save the credentials to our browser, it make it easier to use git commands for later:



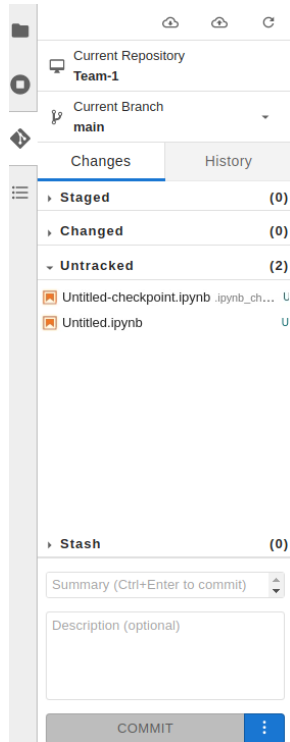
A "Save password?" dialog box with a header area containing a Google Assistant-style character and a cyclist icon. The main area has a "Username" dropdown menu with "lbreguet" selected and a "Password" dropdown menu with dots and an eye icon. Below these are "Never" and "Save" buttons. At the bottom, a message states: "Passwords are saved to [Google Password Manager](#) on this device."

13. The repo has been cloned:

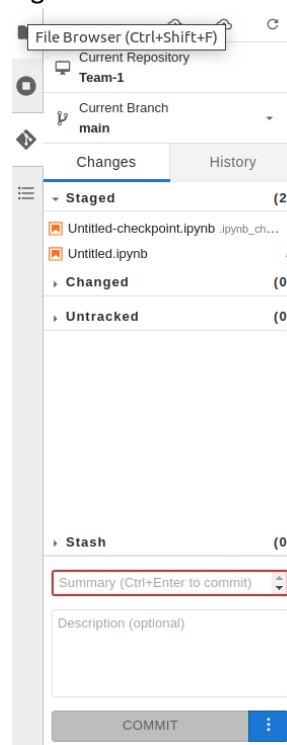
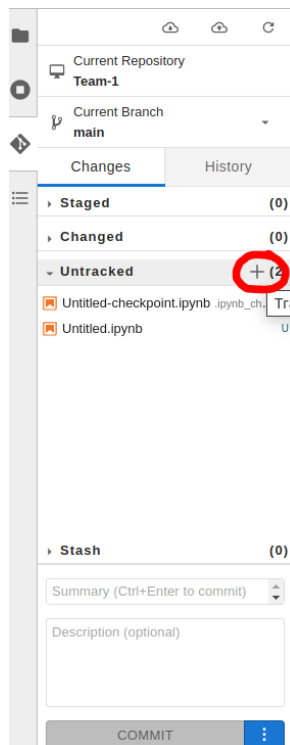
shared_da...	6 months ago
shared_do...	6 months ago
Team-1	2 minutes ago

## B. Committing:




1. After making a change, go to the git icon, you'll notice a version control dashboard:



2. Hover over **Untracked** and press the + sign:




3. Type a summary of your changes at the bottom then press **Commit**:

Stash    (0)

Test Commit

Description (optional)

COMMIT 

4. Input your name and your Github email and press **Ok**:




Who is committing?

Committer name: Louis

Committer email: something@email.com

Cancel OK

5. There will now be an icon with an orange dot indicating to push to your repo:

Current Repository  
Team-1

Current Branch  
main

Changes History

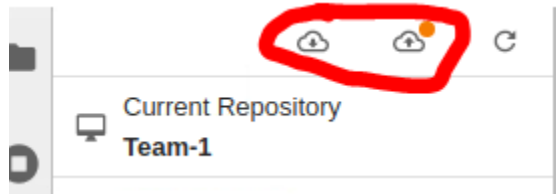
Staged (0)

Changed (0)

Untracked (0)

C. Pushing and Pulling:

1. Periodically, you will have to push your changes and pull your teammates changes from your repo. The cloud with the arrow pointing down is the **git pull** command while the other cloud with the arrow pointing up is the **git push** command:



2. Both pushing and pulling utilize the same UI behavior, so get that Personal Access Token we generated earlier handy. Input your credentials and press **OK**:

### Git credentials required

Enter credentials for remote repository

lbreguet

.....

☐ Save my login temporarily

Cancel

OK

3. You will get a success message on the bottom right if everything was successful:

