Lab 2 Part 2

Gitlab

Overview

The outputs from this lab will be assessed and contribute to your final grade.

In this lab, you will take your Berties Books application from Lab 2 Part 1 and add it to a repository in Gitlab. For those who want a further challenge (for more marks!) there is an extension task later.

You can set up access to Gitlab from your virtual server or from your local computer. This provides a nice simple way to move between the two (e.g. develop and test on your local computer and deploy to the virtual server).

Tasks

Task 1: Log in to your virtual server

Change to your Berties Books directory:

```
cd berties-books
```

Task 2: Check your dynamic web application and add comments to your code

- Run your web application to check whether all routes and pages are working fine
- Tidy up your code and add comments to your main.js and index.js files

Task 3. Install Git version control

Install git:

```
sudo apt-get update
sudo apt-get install git
```

Task 4: Add an SSH key to your GitLab account

You will now create an ssh key to authenticate yourself (i.e. your campus user) on your Gitlab account. If you have previously done this you can skip this task.

Note that you can create an SSH key for each account you use. So you can create one on your virtual server and another on your local computer so you can access GitLab from wherever you are working.

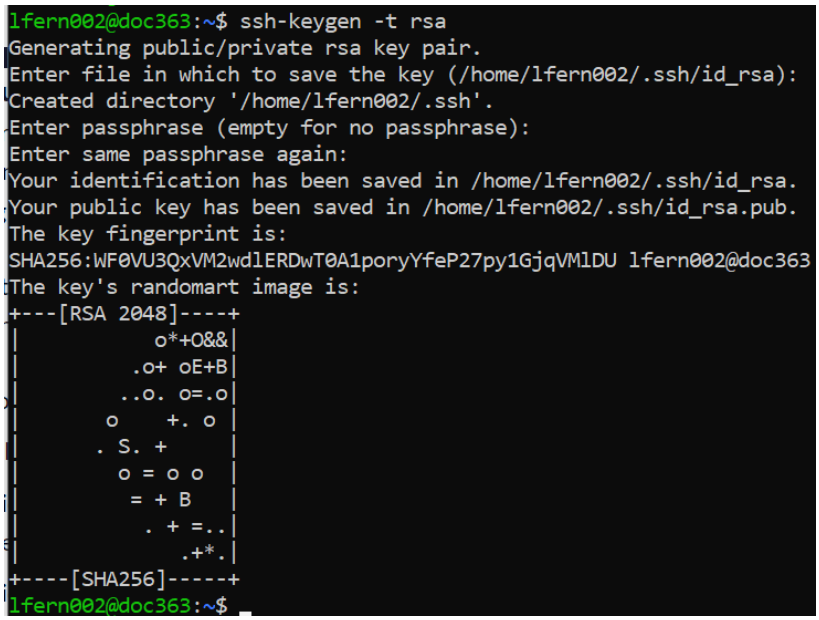
Here I will illustrate the process from my virtual server, but the same steps can be performed on your Windows or Mac computer:

1. Generate an SSH key:

```
ssh-keygen -t rsa
```

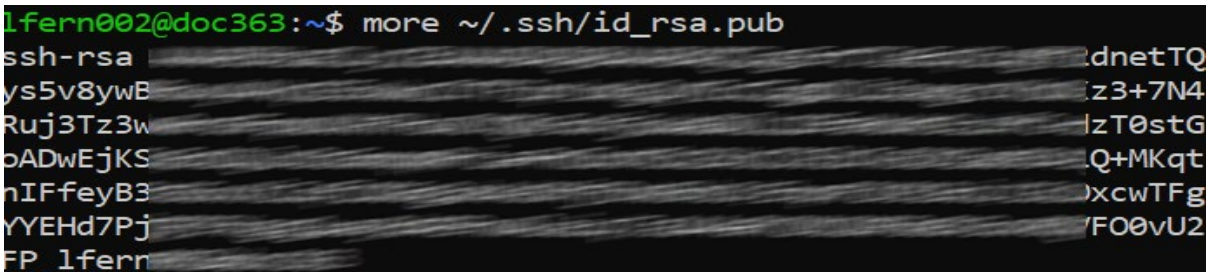
Press <Enter> to accept the default location and file name.

Enter and re-enter **a passphrase** when prompted. **Make sure you can remember it!** It may be easiest to enter the same passphrase you use to login into your campus account. Make sure your passphrase is at least 5 characters.



2. Run the following command to show your key:

```
more ~/.ssh/id_rsa.pub
```



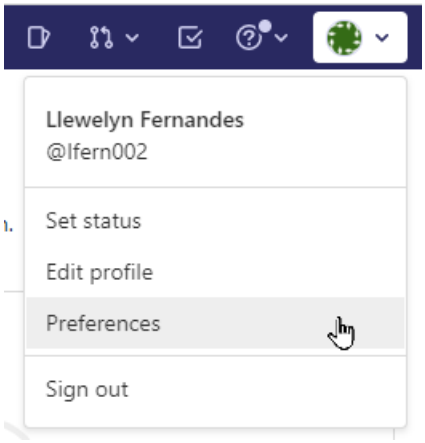
3. Select the contents of the key (all the way from 'ssh-rsa' to the end) and copy it.

4. Log in to GitLab using your campus account:

https://gitlab.doc.gold.ac.uk/users/sign_in

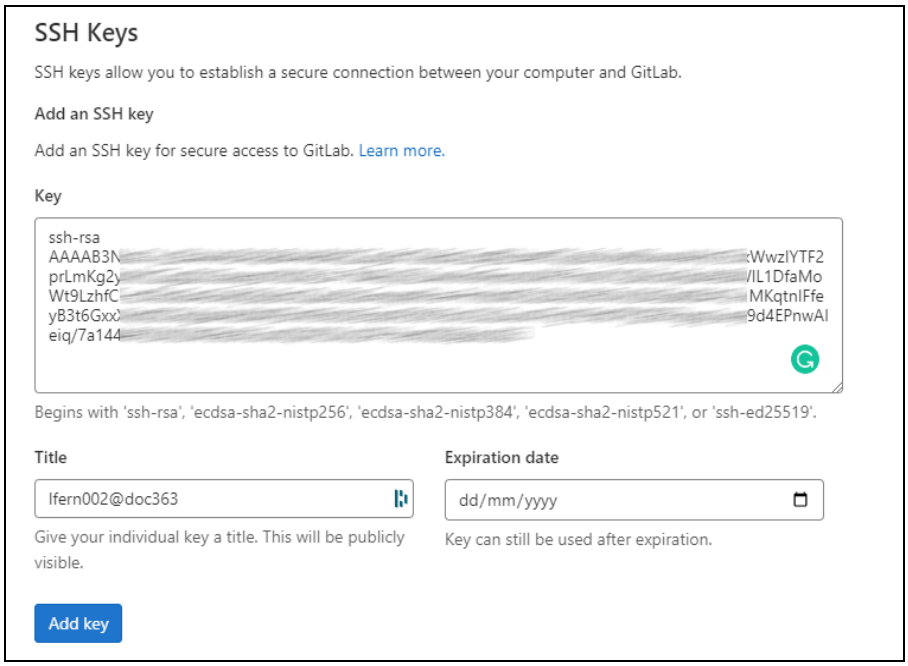
5. In GitLab, navigate to the GitLab home page (dashboard). You get to it by clicking the logo on the top left of the screen.

6. From the top-right drop-down menu, go to **Preferences**:



7. From the left side menu, select, **SSH Keys:**

8. Paste the public key you copied into the box under 'Key'. Note that the 'Learn more' link above will take you to some similar instructions to these, which include help generating key-pairs on Window's machines (useful if you want to access GitLab from a PC rather than on a virtual server).

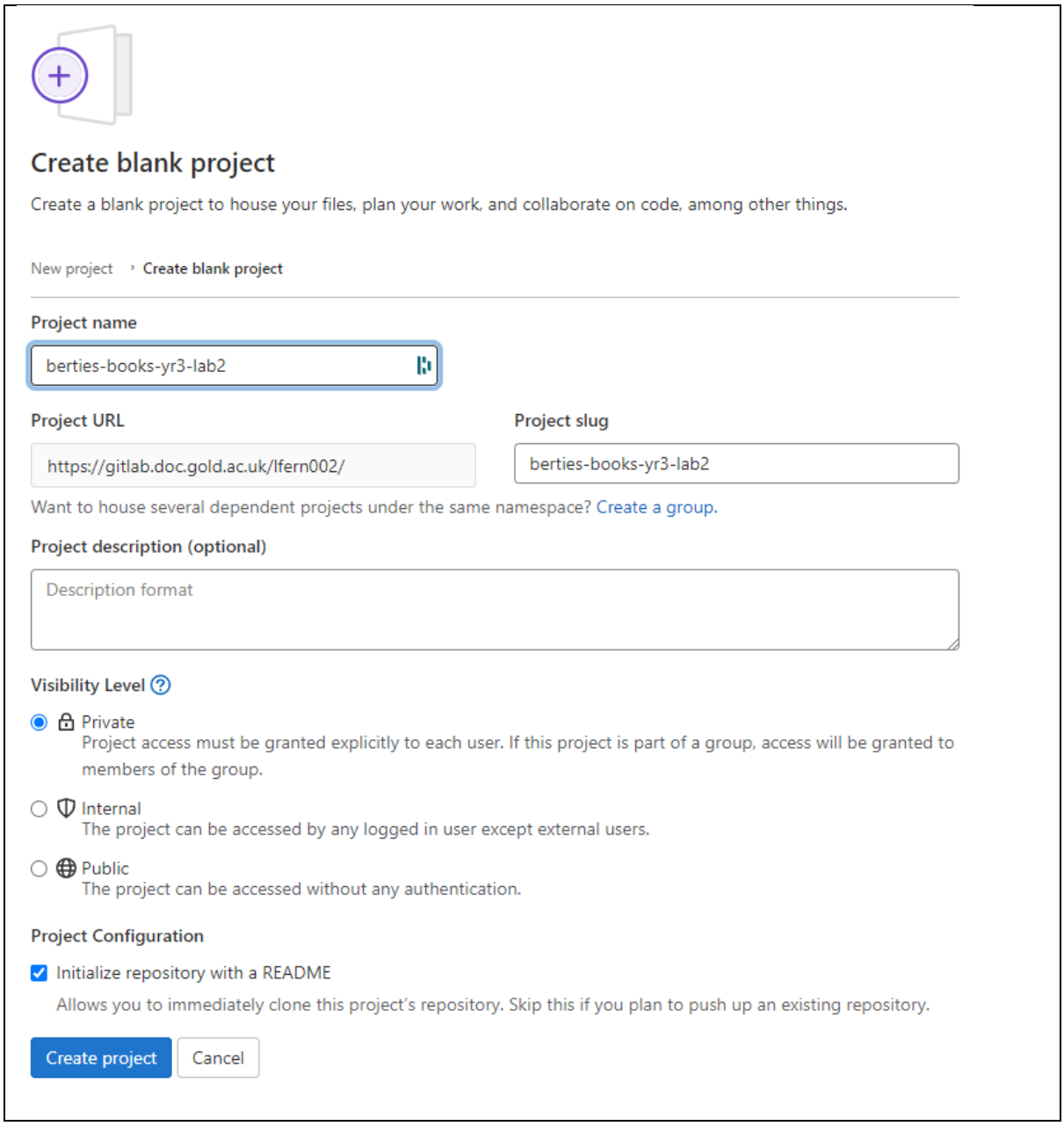


Accept the default title, or enter a suitable alternative such as 'virtual server key' and then click **Add Key**.

Please note that a key-pair is associated with a user account, so if you want to use GitLab from another machine (i.e. as a different user), you will need to repeat these steps from that computer (assuming it is Unix-based). You can share as many public keys as you like with GitLab.

Task 5: Create a git project

9. Go back to the home page in GitLab and create a new blank project named **berties-books-yr3-lab2**. Make sure your project is **private**.



Task 6: Setting git config

10. Set git config in your virtual server terminal by:

```
git config --global user.name "yourname"
git config --global user.email "username@gold.ac.uk"
```

Replace 'yourname' with your name and replace 'username' with your user name.

As an example, this is how I do the git config for myself, please use your name and username instead of mine:

```
git config --global user.name "Llewelyn Fernandes"
git config --global user.email "lfern002@gold.ac.uk"
```

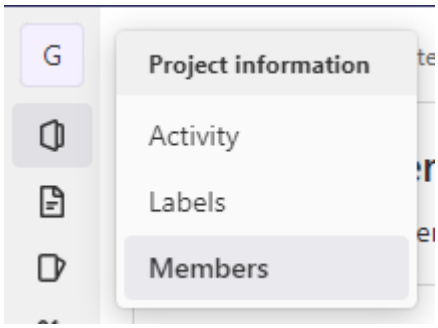
Task 7: Add your project to Gitlab

11. Now, run each line one by one on your virtual server in order to push an existing folder (**berties-books-yr3-lab2** as an example). Replace 'username' with your user name:

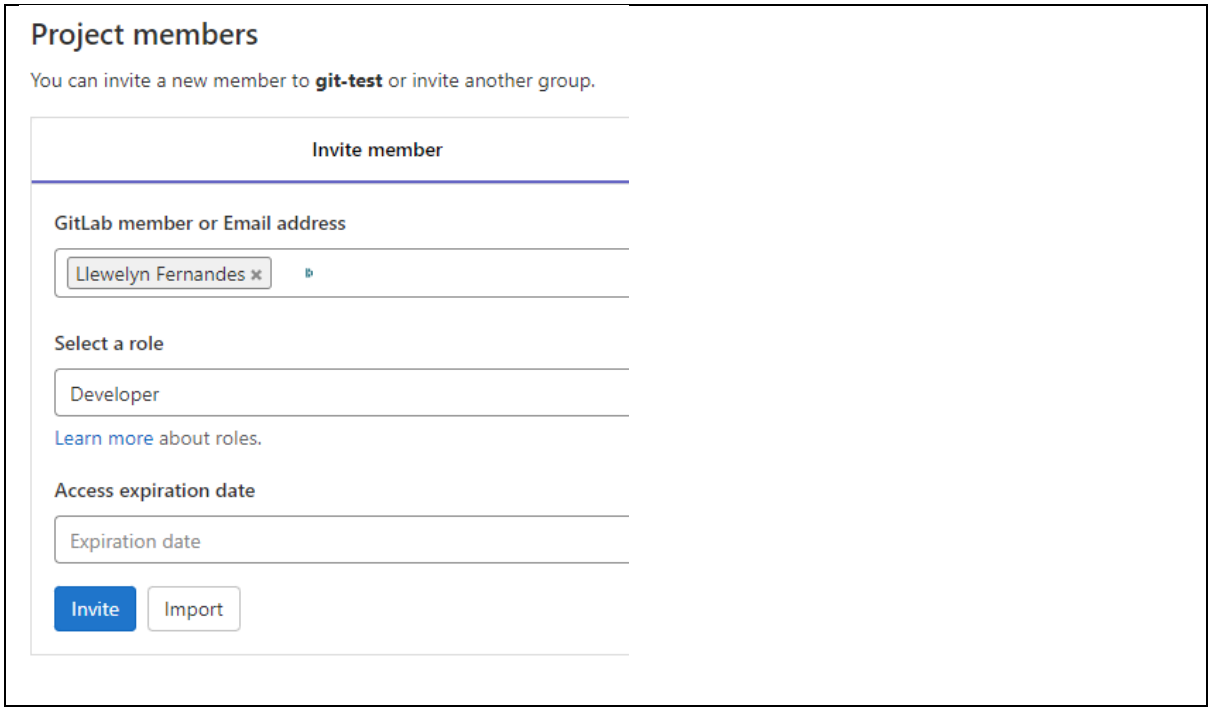
```
cd berties-books-yr3-lab2
git init
echo node_modules >> .gitignore
git remote add origin git@gitlab.doc.gold.ac.uk:username/lab6.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

12. Now check your project page on GitLab and you will see that all files and folders are copied to your project repo.
https://gitlab.doc.gold.ac.uk/users/sign_in

Give permission to me and lab assistants to access your GitLab repo by clicking on the **Project Information** menu on the sidebar and then **Members**



and then giving **Developer** permission to me (**lfern002**) and lab assistants (TBC).



Submit the URL of your project's Gitlab repo on the lab 2 submission page.

Task 8: Extension

This is an extension task for this lab and is worth 10% of the mark.

Add a new page (route) to your dynamic web application called `/deleteuser` to delete a registered user from the database. This page should display a form to collect a username and then delete all data related to that user. You can check if the user's record is deleted by checking `listusers` after the delete operation.

Task 9: Pushing subsequent changes to Gitlab

If you make any further changes to your code after the first push (e.g. if you do the extension task above) you will need to push them to Gitlab. You can follow the instruction below to do this.

If you have not made any further changes to your code after the first push, skip this task for now (but note how to do this as you will need it in the future!).

Run each of the following lines, one at a time in order to push the code changes to your GitLab project (i.e. local repo to remote repo):

```
cd lab6
git add .
git commit -m "second commit"
git push -u origin master
```

Try to use a sensible comment for the commit that describes the changes made (i.e. don't use "second commit" like I have!).

Task 10: Submit TWO URLs

After successful completion of all tasks in lab 2 part 1 and 2, your app is ready to be marked.
Remember to give permission to me and lab assistants to access your GitLab repo.

Then run your web application with **'forever'** and submit TWO URLs on the lab 2 assessment page:

1. URL of your git repo,
   example: https://gitlab.doc.gold.ac.uk/username/projectname
2. URL of your web server: www.doc.gold.ac.uk/usr/ID/

YOU WILL ONLY BE MARKED IF BOTH URLs ARE SUBMITTED and working.

The marking scheme is as follows:

- Register and Registered pages (with hashed passwords): 10 marks
- Navigation from all page to home page and vice versa: 10 marks
- Nice, tidy code with comments: 10 marks
- Listusers page 10 marks
- Deleteuser page (extension task 1, see above) 10 marks
- Login and loggedin pages: 50 marks

Please refer to the lab 2 rubrics for more information on marking.

END