Project 1: HTML+CSS

2019-05-06 22:54
1a0f14a

Contents

For this assignment, you will design and implement a small static web site using hand-coded HTML and CSS.

Requirements

- Your site should have **three** separate HTML pages that are each linked to all the others. The main (home) page should be named `index.html`.
- Use an external style sheet called `style.css` so that the style rules for all three pages are consistent.
- Among the three pages, there should be at least **four** images.
- At least one page should use a multiple-column layout.

Tool suggestions

- Use **Atom** as your code editor. In the VM, it is accessible as ⚙ » **Development** » **Atom.**
- Once you have a `.html` file saved in Atom, you should try out **Packages** » **Preview HTML** » **Enable Preview**.
- Helpful key shortcut: **Ctrl-Alt-period** will close the currently-open tag.
- Also try **Edit** » **Lines** » **Auto Indent**.

How to submit

If you are using the VM for both courses this semester (CS120 and CS164), you only need to do this first subsection **once.** Repeat starting at the subsequent section **(Create gitlab project)** for the other course.

Continue git setup

1. In your VM, open ⚙ » **Terminal Emulator** and type the following two commands, replacing the upper-case portions in double quotes with your **actual name and email address.** (You should use the same email address that is registered on `gitlab.liu.edu`.)

   ```
   git config --global user.name ”YOUR NAME”
   git config --global user.email ”YOUR.ADDRESS@EXAMPLE.COM”
   ```

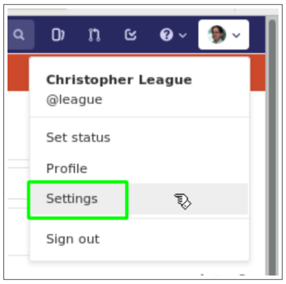
(If it says “git: command not found,” you need to update your VM configuration – just type `liucs-update`.)

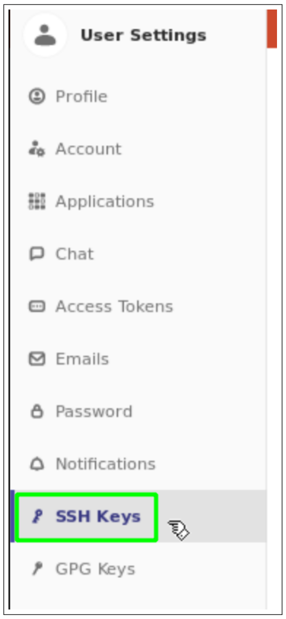
2. Next, generate an SSH key pair by typing `ssh-keygen`. You’ll **press enter** for the defaults, **including an empty passphrase.** It should go something like the following, though the fingerprint and “random art” will be different, and the ↩ symbol represents where you press enter:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa): ↩
Enter passphrase (empty for no passphrase): ↩
Enter same passphrase again: ↩
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:WSZdJ850SVsi6WbOHyI8j2l4b3YA96HyWqXSnJx3AtA student@liucs
The key's randomart image is:
+---[RSA 2048]----+
|            =o=..|
|          . *.=.+ |
|         . =.E .  |
|          =..= .  |
|         S. B.o.. |
|           =+**o  |
|          ..OO+o..|
|         . =o=.oo |
|          o.=..   |
+----[SHA256]-----+
```

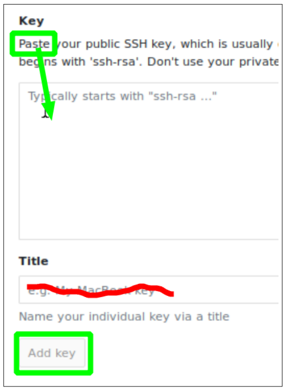
3. Type `atom ~/.ssh/id_rsa.pub` and wait for Atom to pop up. It should show some text that begins with `ssh-rsa AAA...`. Use **Ctrl-A** then **Ctrl-C** to copy the entire code onto the system clipboard.

4. Use ⚙ » **Web Browser** to log in to your account at `gitlab.liu.edu`. Select **Settings** from the profile menu in the upper right, then **SSH Keys** from the left sidebar, as shown in the screenshots.





5. Paste the `ssh-rsa` code into the big box as indicated, then hit **Add key.** You don't need to provide the Title.



Paste your public SSH key, which is usually begins with 'ssh-rsa'. Don't use your private

Typically starts with "ssh-rsa …"

~~e.g. My MacBook key~~

Name your individual key via a title

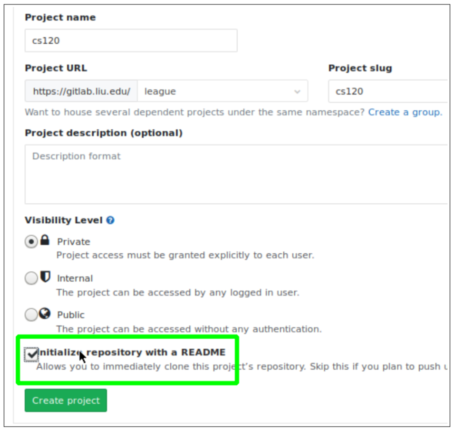
Create gitlab project

1. Still on `gitlab.liu.edu`, use the upper left menu to go back to your projects list. Click the green **New Project** button.

2. In the **Project name** box, enter "cs120".

3. The **Project URL** and **slug** should be fine as they are, and you don't need a **Project description.** Keep the **Visibility level** set to **Private.**

4. Select the checkbox to **Initialize repository with a README.**



Project name

cs120

Project URL https://gitlab.liu.edu/ league

Project slug cs120

Want to house several dependent projects under the same namespace? Create a group.

Project description (optional)

Description format

Visibility Level

● Private
Project access must be granted explicitly to each user.

○ Internal
The project can be accessed by any logged in user.

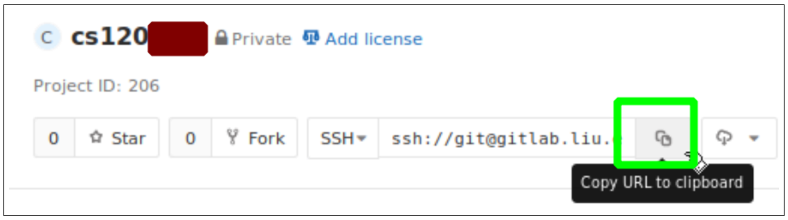
○ Public
The project can be accessed without any authentication.

☑ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push u

5. Hit **Create project.**

6. On the page for your new project, look for **Settings » Members** in the left sidebar. In the **Search for members** box, type `league` and select the user named exactly `@league`, with my picture. Change the **role permission** drop-down to **Developer** and then click the green **Add to project** button.

7. Go back to the main project page (cs120 at the top of the left sidebar), then find the `ssh://` URL and use **Copy to clipboard.**



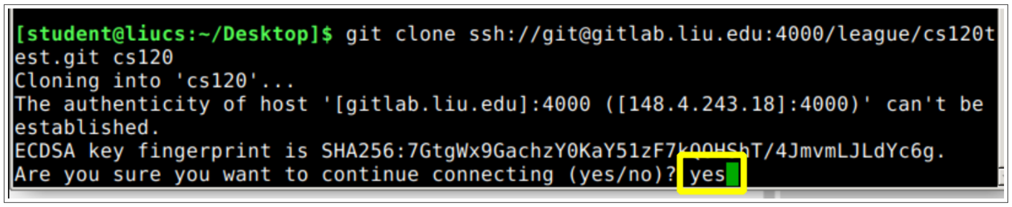
cs120 [redacted] Private Add license

Project ID: 206

8. Back in the **terminal,** change to your `Desktop` directory, and then run the git command. You can paste the URL instead of typing it by right-clicking or using Shift-Ctrl-V:

```
cd ~/Desktop
git clone ssh://git@gitlab.liu.edu:4000/USERNAME/COURSE.git
```

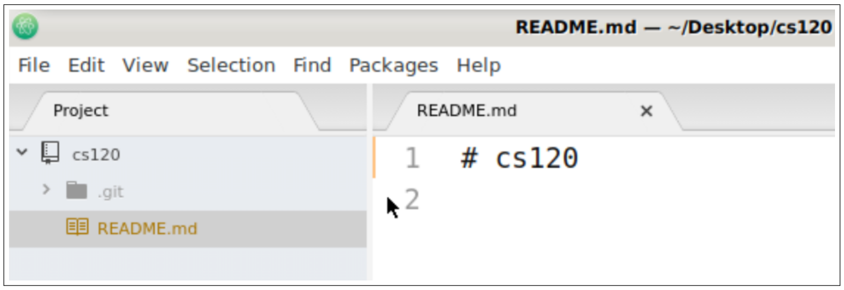
The first time you connect from a new VM, it will probably print a fingerprint and ask whether to continue connecting — type out `yes`.



```
[student@liucs:~/Desktop]$ git clone ssh://git@gitlab.liu.edu:4000/league/cs120t
est.git cs120
Cloning into 'cs120'...
The authenticity of host '[gitlab.liu.edu]:4000 ([148.4.243.18]:4000)' can't be
established.
ECDSA key fingerprint is SHA256:7GtgWx9GachzY0KaY51zF7kQ0HSbT/4JmvmLJLdYc6g.
Are you sure you want to continue connecting (yes/no)? yes
```

Commit a change from VM

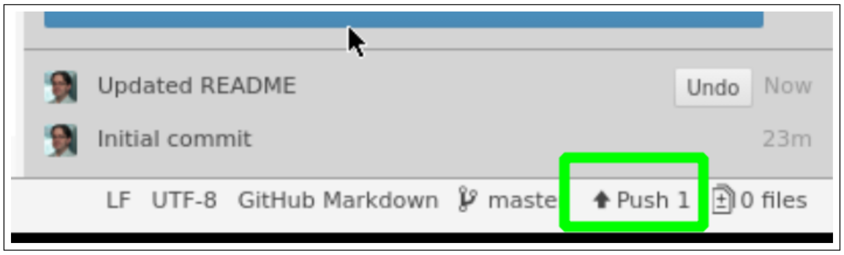
1. Switch back to **Atom,** then use **File** » **Open Folder** and navigate to **student** » **Desktop** » **cs120**.
2. Click `README.md` in the **Project** file tree to open it.



```
# cs120

```

3. Make a small change to the `README.md` file, perhaps by typing your name into it. Save the change.
4. Select **View** » **Toggle Git Tab** or Ctrl-Shift-9.
5. Hit **Stage all** in the upper right.
6. Type `updated README` into the box labeled **Commit message** and then hit **Commit to master.**
7. The commit should appear in the history list in the lower right, just above "Initial commit." (They are in reverse chronological order.)



8. Hit **Push** to upload your changes to the gitlab server.

Submit your project

1. Now you are finally ready to submit your project file(s), and all future projects. (We'll use the same GitLab project repository throughout the semester — don't create additional projects unless I advise you to.)
2. Use ⚙ » **File Manager** to move your project file(s) into **Desktop** » **cs120**.
3. Go back to the Atom **Git Tab**. It should notice the new file(s) under **Unstaged** changes. Repeat the same **commit** and **push** steps as before. You can use "Submitting project 1" or similar as your commit message.

4. Refresh your project page on `gitlab.liu.edu` and you should see evidence of the submission.