**Website Tutorial**

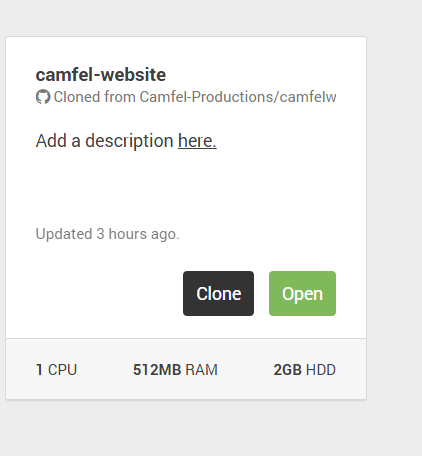
**Login to C9**

Go to <https://c9.io/login>

Enter in the following username and password

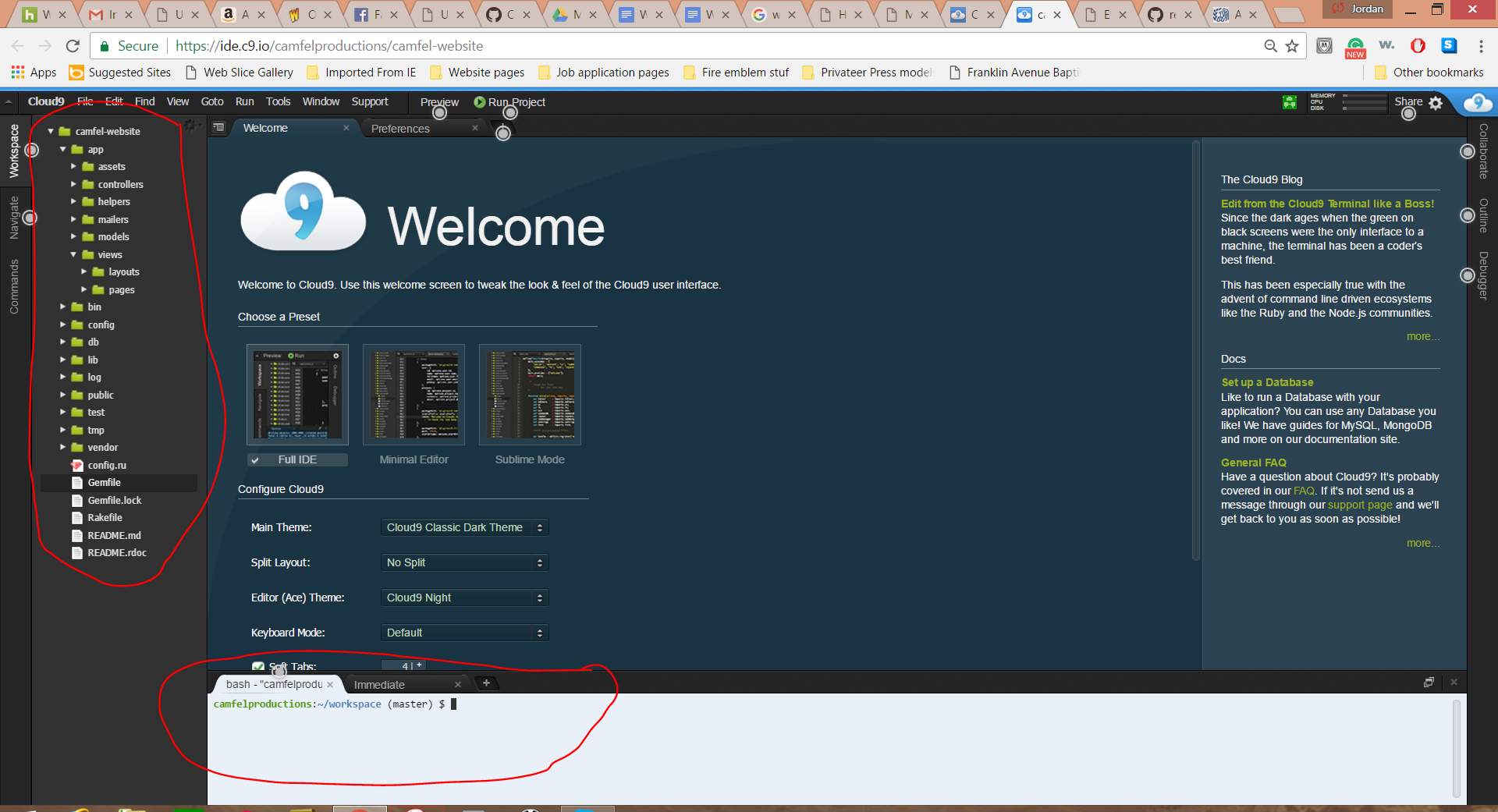
Username: [camfel2017@gmail.com](mailto:camfel2017@gmail.com)

Password: Romans55



Click the “open” button

which opens a new tab with the workspace.



Files can be accessed through the hierarchy on the left hand side (circled). Console commands can be accessed through the console window at the bottom (also circled).

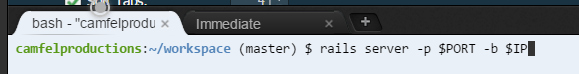
**STEP 1: Run test view**

The first thing to do is to set up a test area to make sure the changes made don’t break the system, allow you to roll back changes and look the way you want it to. To do this, go to the console window (circled in the photo above) type in the command below and hit enter.

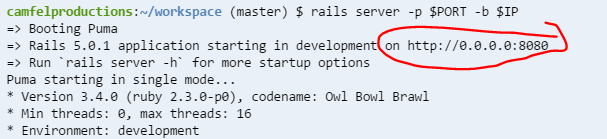
To type in: git checkout -b test



To type in: rails server -p $PORT -b $IP



**Once you hit enter, click on the circled link below**



Open that and you should have the website in a different tab. This way you can see what the changes look like. When finished with the test server, it can be cancelled by executing the command: ctrl C

**Optional Step: Oops I screwed up**

If you break something and need to revert, execute these following commands.

To type in: git checkout master



This command switches branches to the system that works. You should see it look like that when finished. The next command deletes the broken branch you were working on (called test).

To type in: git branch -d test



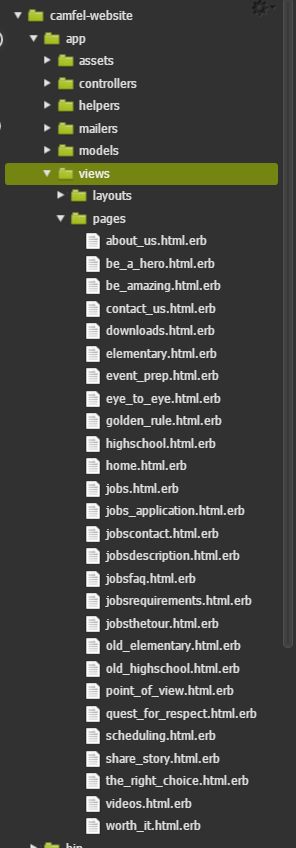
Now that it’s deleted, you can reuse the command: *git checkout -b test*  to create a new branch to work in. the circled portion tells you what branch you are currently in.

To type in: git checkout -b test

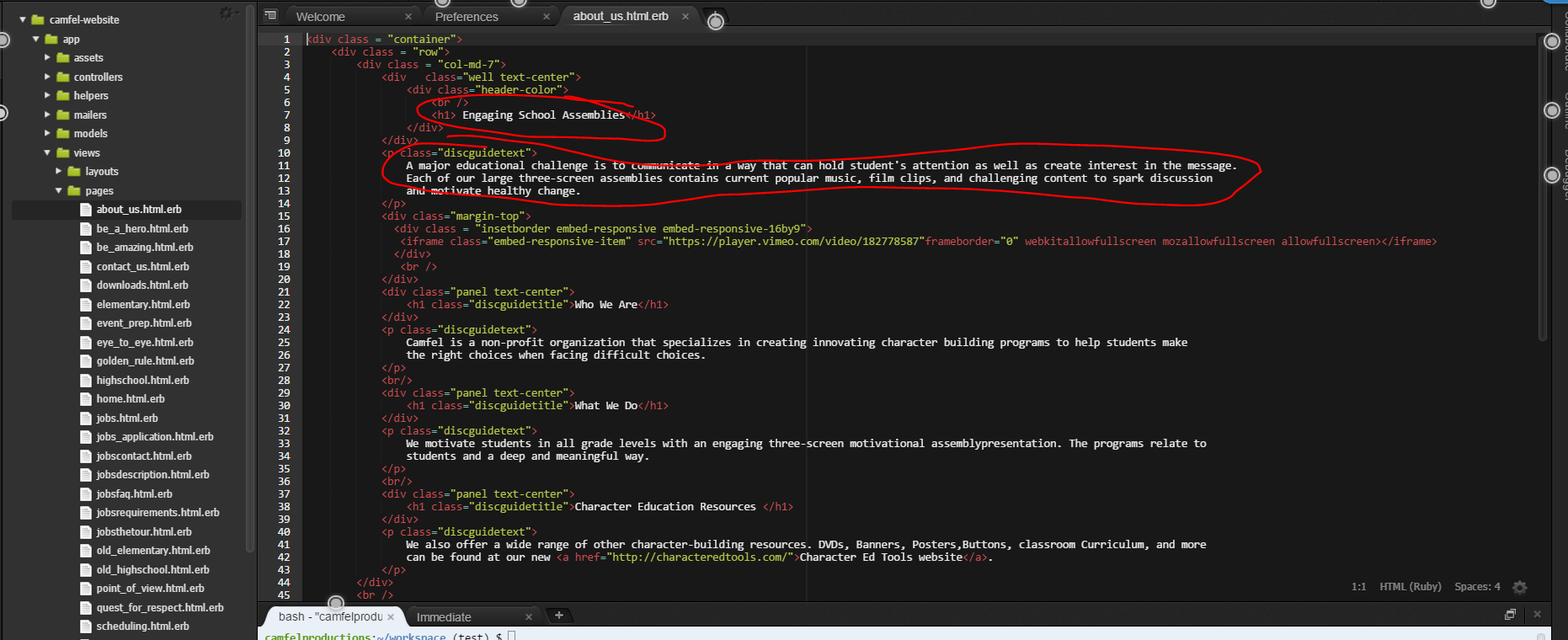


**Step 2: Changing text**

To edit the text the first thing a person needs to do is to find the html.erb files. Those files are the ones with the text and the structure of the website. They are in app/views/pages as shown below.

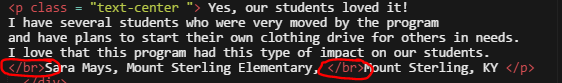


The file name corresponds to the name of the page which you are trying to change. Open one by double clicking on it and you should see it as shown below.



Any white text on the page (Two examples are circled) can be edited and altered. The text will fit in the format and line breaks are placed according to that.

Text and line breaks look like this:



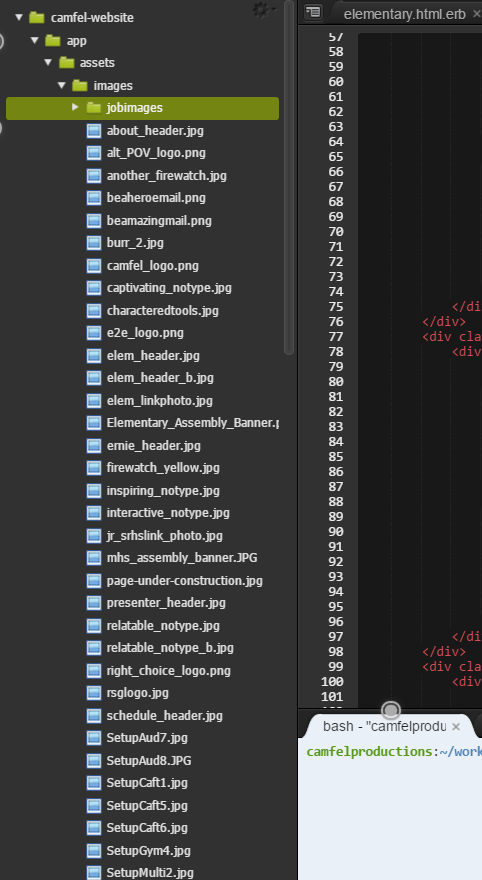
By typing in the code: <br /> a line break can be placed, if the default line break isn’t satisfactory.

Simpler Instructions: Find white text similar to the ones shown above. Type in whatever you want.

**Step 3: Changing pictures**

To edit the text the first thing a person needs to do is to find the picture files. They are in app/assets/images as shown below.

Drag pictures from computer here.



To add a picture to the website, simply drag a photo from your computer’s file system over to this and it should add automatically. Make sure the path for the image stays inside the images folder (meaning it joins the many other images shown here in alphabetical order).

Once you’ve added the image that you want, you need to find the image you want to replace. Images, like text, are found primarily in the html code.

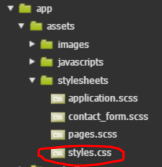
Images look like this:



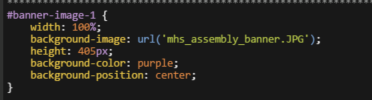
To edit, change: “jr\_srhslink\_photo.jpg”

In the image tag you can see the name of a file surrounded by quotation marks. All that is required is that you put the new image name in the old one’s place.

If you can’t find a picture’s reference in the html, first find it in the hierarchy so that you know the name of it. Then once you know the name of it, go to the css file. (called styles.css)



Once there, search for the image and replace it by changing the name to the new image you want. Below is an example image:



In this example, the only thing to change is “mhs\_assembly\_banner.JPG”. change this to the new image. For example it would say “url(‘new\_image.jpg’);”

Simpler instructions: Drag image from computer to section with images. Find the green text which is a file name for a picture. Change the green text to the name of the file you want.

**Step 4: Changing links**

There are two types of links. Links that go between pages on your site, and links that go to other sites.

Links look like this:



To Edit, Change: eye\_to\_eye\_path but keep \_path at the end

In this example, the text “Option #1” tells it what the text should look like. This text can be altered to anything you would prefer with no consequences. “eye\_to\_eye\_path” tells it where to send the user. To edit this change “eye\_to\_eye” to whatever your new page is called. So if you want it to go to “the\_new\_show.html.erb” you would change the link to “the\_new\_show\_path”.

Some links look like this:

****

This link is very similar to the other one. The way to change where it links to is to change the portion that says “the\_right\_choice\_path” to the desired path.

To Edit, Change: the\_right\_choice\_path but keep \_path at the end

Links that go outside your site might look like this:



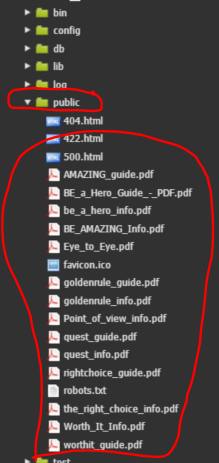
To change where these go, you simply put the new url in place of the old one. This url is “http://characteredtools.com/assemblies/questassembly.html”.

To Edit, Change: “http://characteredtools.com/assemblies/questassembly.html”

Simpler instructions: find the words that have underscores instead of spaces. Notice they’re the title of the page they go to. Change the words to the title of the page you want it to go to.

**Step 5: changing PDFS and other files**

To add a new discussion guide go to the folder called public. Shown below.

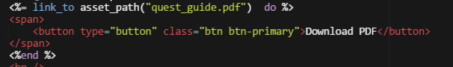


Drag the new pdf into this section.

This is also where new flyers get placed.

The pdf discussion guides are referenced in two places. On the actual page, and also in a link.

This is the link to the pdf download.



To change pdf, change: “quest\_guide.pdf”

Change what pdf is being linked to by changing the words “quest\_guide.pdf” to whatever the new pdf is called. For example “new\_show\_guide.pdf”.

This is the PDF on the actual page:



To edit, change: “/quest\_guide.pdf”

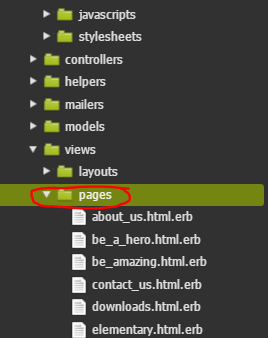
To change this, change the words “/quest\_guide.pdf” to whatever you like, keeping the slash. So for example changing it to “/new\_show\_guide.pdf”

Simpler instructions: Drag pdf to files of other pdfs. Find green text ending in “.pdf”. Change file name to new file name.

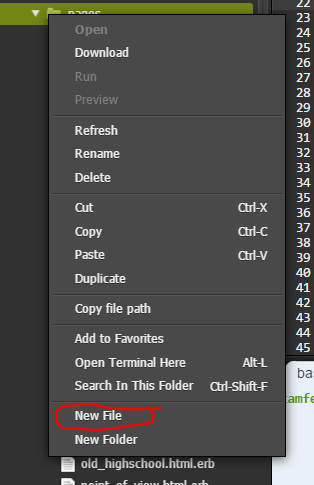
**Step 6: Adding a Page**

When it’s time to add a new show page, this is how it works. First you go to the hierarchy and right click on the “pages” folder.

To do: right click folder named “pages”



Once you’ve done that, you choose “new file”



Call your new file whatever you want the page to be named, followed by “.html.erb” for example: “the\_new\_show.html.erb” as shown.

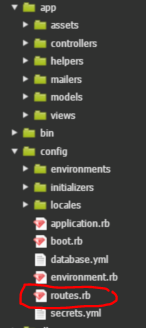
Name file like this:



To do: open the new file. copy the quest for respect page. Change text, images, links, etc to what you want.

Once that is done, open it by double clicking it, and open another show page and copy all the old page’s code into your new show’s page. Then edit all the text and pictures so that they are related to the new show instead of the old one. To go along with this, I included instructions in the quest for respect page. All the gray text is a comment showing what can be changed. The gray text is not exhaustive, but rather a few examples. Once you are happy with the page, it is time to add it to the file paths. To do this there are two files it needs to be in.

To Do: open routes.rb



The first file, as shown above is routes.rb. It is found in the “config” folder.

to add a page to routes.rb you follow the syntax of the existing pages, as shown below.

Copy this:



To Do: change words “jobs\_application” to your file name.

The above reference is for a page called “jobs\_application”. Simply change everywhere it says jobs\_application to your new title.

If your new page was called “the\_new\_show.html.erb” this reference would look like:

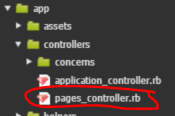
Example of “the\_new\_show”:



Now you are done with the routes.rb file.

The next file we need is the pages controller.

To do: Open pages\_controller.rb



Changing this one is also very similar. An example is shown below.

Copy this:



To Do: change words “jobsfaq” to your file name.

**Step 7: Pushing to github**

If you are happy with your changes, **(Look at the test)** it is time to push the changes to github. (our version control software) the first thing to do is execute the command “git status”

To Type In: git status



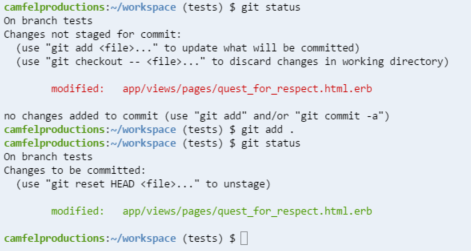
This will show you all the files you changed. They are in red because they haven’t been backed up yet. The next command is “git add . “ This prepares all the files changed to be backed up.

To Type In: git add .



To Type In: git status

To notice: files which were listed in red, are now green



The next step is to switch back to the master branch. To do this execute the command “git checkout master”

To Type In: git checkout master



Then execute the command “git merge test” to merge the branch with the master.

To Type In: git merge test



I called mine tests instead of test. The next step is to say “git commit -m “message” “

To Type In: git commit -m “a comment of what you did”



The message is for you, so you can see what was changed, type whatever you want. It cannot be done with an empty message though. Below it you’ll see how many files were changed and how they were changed.

Notice that it says files have been changed.



The next thing to do is push it to github. This is done by executing the command “git push”

To Type In: git push



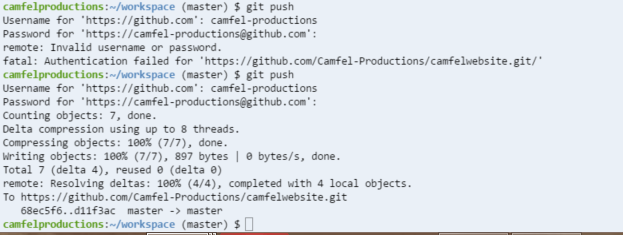
Username: camfel-productions

Password: romans55

Notice: It doesn’t show you that keys are being pressed for the password, but it is recording them.

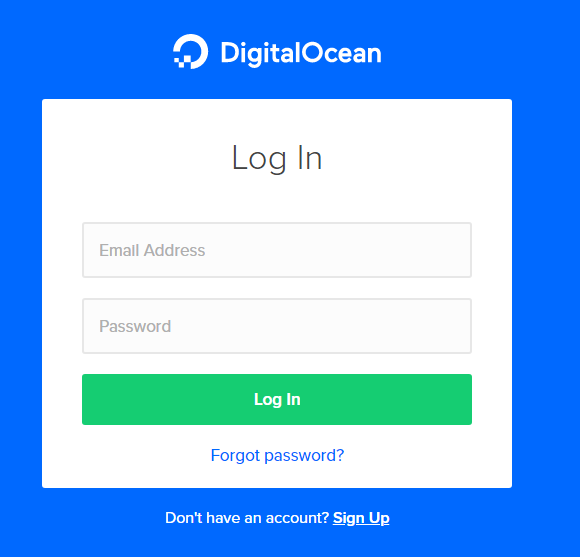
It’ll ask for the username, which is camfel-productions. And the password, which is romans55

This is what it looks like to fail, and then succeed.



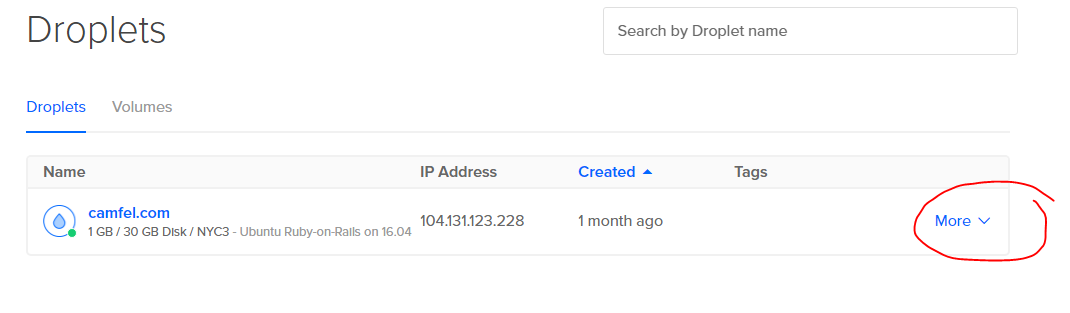
**Step 8: Pushing to server (DigitalOcean console)**

Go to <https://cloud.digitalocean.com/login>

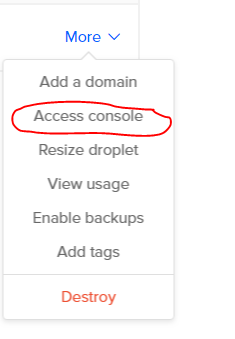


Type in: [jordan@camfel.com](mailto:jordanzeeg@gmail.com)

Password: Amazing1!



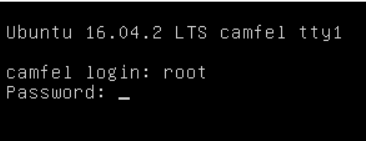
Go to more, go to access console



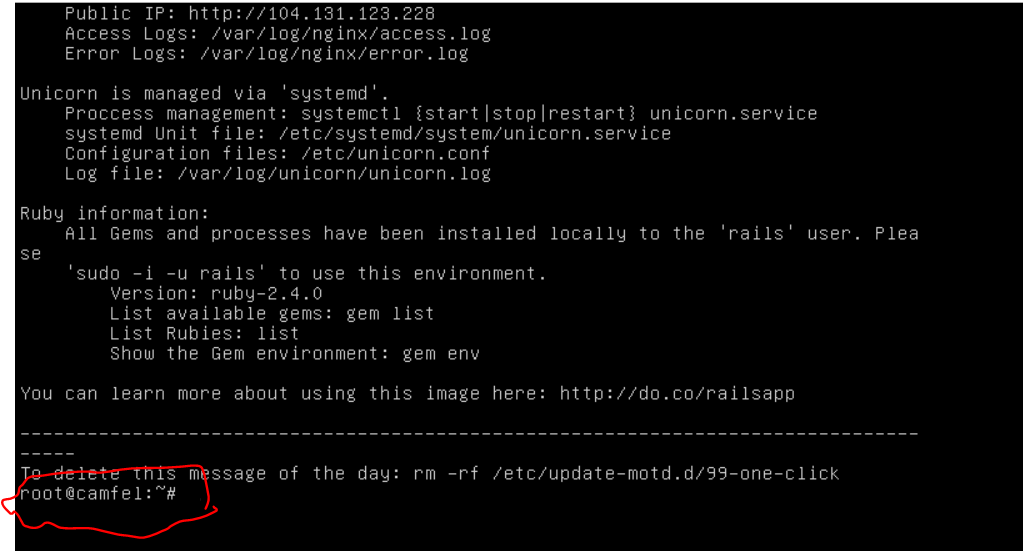
If it asks for camfel login:

Camfel login: root

Password: Amazing1!



Itll then say **“root@camfel:”**

****

Then type in: sudo -i -u rails



Type in: ls

This will show you all the files in the current area.

Type in: cd camfelwebsite

Cd stands for change directory.

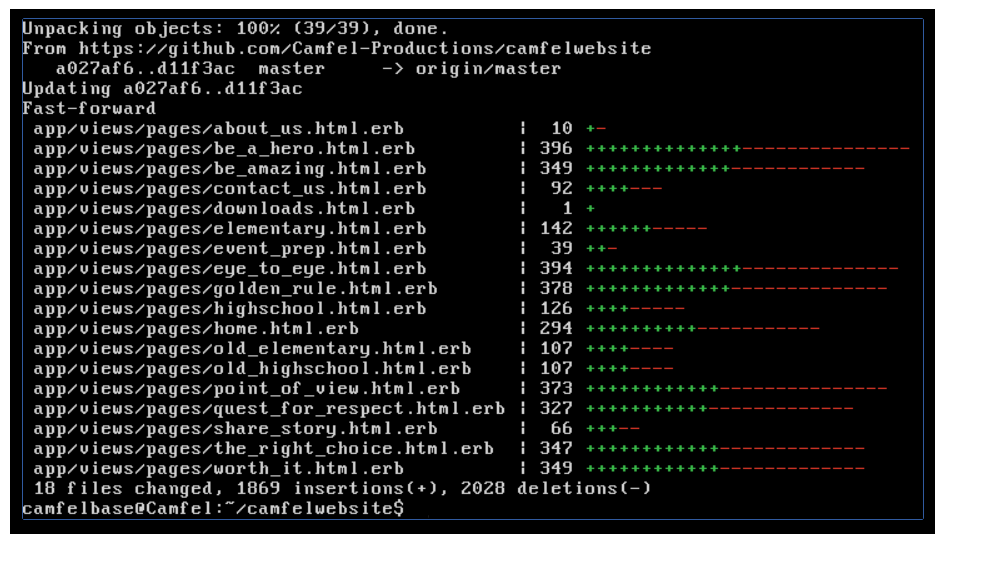
Type in: git pull

It should ask for the github username and password.

Username: Camfel-Productions

Password: romans55

Results similar to this show that you succeeded.

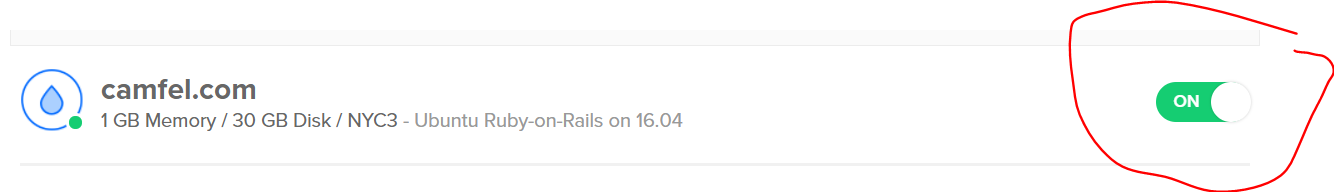


Log out of rails user using command “logout”



Then power down the server using command “poweroff”

Then turn the server back on using button on digitalocean.



Give the server a moment to turn on, and now it should all be live.