**This is to help with the deployment of the timetracker app**

Hello! If you are reading this, it means you are currently working on the Time Tracker app! I will go over what I was working on at the time of the handoff.

I was working on getting the app deployed to the student server: **IP Address: 137.190.18.16**

It is a **linux** server running **ubuntu**. In order to gain access to it, you will

1. Get permissions given to you from Brad
2. Use your terminal or command line and log in
   1. For terminal use: ssh <<username@137.190.18.16>> then enter your password
      1. For example I would type: ssh tuckerbrady@137.190.18.16
      2. Then type in my password
   2. For command line, not really sure, but I would imagine it’s very similar to ‘a’
3. Once you have entered the server, you can Change Directory into the timetracker folder
   1. Use the command: cd timetracker
4. Once you are inside the timetracker folder, you will need to make sure you navigate to the correct branch of the git repo. (docker-deploy)
   1. Use the command: **git branch** to see what branch you are currently on
   2. Use the command: **git checkout docker-deploy** to move to the correct branch
   3. Use the command: **docker-compose up** to get the app up and running
      1. You should see the database\_1 boot up
      2. You should see web\_1 boot up
      3. If this doesn’t happen, you may need to build the app again
         1. Use the command: docker-compose build
      4. At any time you can see what docker images are running
         1. Use the command: docker ps
         2. Or use the command: docker ps -a
   4. Continue to use the docker-deploy branch until you get the reverse proxy up and running
   5. To shut down the app run the command: hold the control button and press ‘c’
5. Here is a link to the last tutorial I was trying. You should see the coordinating code in the ‘Dockerfile’ and ‘default.conf’ files in the program (I’ve left comments there as well)
   1. <http://littlebigextra.com/install-nginx-reverse-proxy-server-docker/>
   2. I have already created the nginx base image (terminal)
   3. I have already created the custom nginx image (default.conf)
   4. I have added the code for the docker image (commented out in Dockerfile)
      1. I was starting to run into an issue here when I tried to run: docker-compose up in the server, it might have to do with the next step that I didn’t entirely get around to
   5. YOU will need to work on ‘Running your own Custom Nginx Image’ step.
   6. If everything works as intended –lol—you should be able to see you app up and running on the server!
   7. There is a corresponding YouTube video for this tutorial as well
6. You will be working entirely in the command line/terminal for this section, get used to it, don’t be afraid. Here is a quick little cheat sheet you might find useful
   1. <https://gist.github.com/cferdinandi/ef665330286fd5d7127d>
7. If you want to run the app locally from the docker-deploy branch, you need to
   1. Make sure you are not in the linux server
   2. In the terminal use the command: docker-compose up
   3. You will see the app boot up as described above
   4. In your web browser go to localhost:52082
   5. To shut down the app run the command: hold the control button and press ‘c’
8. Good Luck!