Important announcement

The rest of the course will be using the Flask application as an example due to how little changes between deploying a Flask and Rails application. Don't worry though, I will include a fully working version of everything for Rails at the very end of the production server section.

Staging server - Did you follow along with Rails?

If you followed along with Rails then you need to do a little bit of housekeeping inside of xubuntu.

If you followed along with Flask and skipped Rails entirely, then you can ignore the rest of this document and continue on. Otherwise, read on!

The issue is, for both Flask and Rails apps I decided to use mobydock as the Docker image name. You will get a conflict due to this if you follow along without following this 4 step text lecture.

4 steps to fix conflicts

Step 1) Remove the old Docker volumes

```
sudo rm -rf ~/.docker-volumes
```

Step 2) Force delete the Rails based mobydock_mobydock image

```
docker images
docker rmi -f <image id of mobydock_mobydock>
docker rm $(docker ps -q -f status=exited)
```

Step 3) Back up the Rails app if you want to (this course will not reference it again)

```
# Zip up the ~/Projects/MobyDock/mobydock folder and save the zip somewhere safe, perhaps in the ~/Projects/MobyDock folder with the filename of mobydock-rails.zip.
```

Step 4) Implement the Flask app in 1 of 2 ways

```
# If you have a backup of it from before, unzip and use that. If you do not have a backup, then go back and follow along with the entire Flask app section.
```

```
[you may now proceed with the rest of the staging server section]
```