Production server - Preparing your droplet

[download + unzip the Flask resource associated to the video lecture]

Make the deploy.sh script executable again

```
chmod +x deploy.sh
```

Check out the hostname on the staging server

```
ssh you@192.168.1.99 hostname
```

<Replace you with your username, or omit this if it's the same>

<Replace 192.168.1.99 with your server's IP if it is different>

```
[
  replace 192.168.1.99 in the following files if that is not your IP:
   -  deploy.sh
   -  nginx/docker-entrypoint
   -  nginx/configs/default.conf
]
```

[replace STAGING_HOSTNAME in nginx/docker-entrypoint if it differs]

[replace the nick user in units/mobydock.service (2 spots)]

Generate a secure token for the Flask setting SECRET_KEY

```
python
```

[watch the video lecture to see how to generate it]

[watch the video lecture to see how to change your database password]

Move into the deploy folder

```
cd ~/Projects/MobyDock/deploy
```

Pre-seed the production server

```
APP_ENV="production" SERVER_IP="x.x.x.x" SSH_USER="root"
KEY_USER="$(whoami)" ./deploy.sh -P
<Replace x.x.x.x with your droplet's IP address>
<Enter yes to continue connecting>
```

Provision the production server

```
APP_ENV="production" SERVER_IP="x.x.x.x" SSH_USER="root"
KEY_USER="$(whoami)" ./deploy.sh -a
<Replace x.x.x.x with your droplet's IP address>
```

```
[open a 2nd terminal tab by pressing CTRL+SHIFT+T]
```

SSH into the production server

```
ssh you@x.x.x.x
<Replace you with your username, or omit this if it's the same>
<Replace x.x.x.x with your droplet's IP address>
```

Homework assignment

There's an issue with the deploy.sh script. On the first bootup of the machine, the iptables rules are being executed before Docker is loaded but at this point right now Docker is already loaded.

Line 166 of the deploy.sh script attempts to remedy this situation by restarting Docker. Typically this would work, because Docker would restart and append to the iptables rules.

However in our case, the restart of Docker on line 166 doesn't seem to have any effect.

We need to manually fix this by running this directly on the production server:

```
sudo systemctl restart docker
```

This issue only occurs on a fresh provision of the server. If you reboot the machine completely it will fix itself because the iptables unit file will get executed before Docker and all is well.

Your assignment is to refactor the deploy.sh script so that the iptables configuration happens before Docker is even installed so that this bug doesn't even happen in the first place.

This is not a terribly large refactor but you will need to ensure the iptables unit file is copied over separately from the other unit files, this way you can enable it with systemd.

This is an ungraded homework assignment but it's worth doing for sure. I will not be including a working example of the refactor because I'm a huge advocate of learning by doing.

Good luck and if you choose to partake in this mission, I will be interested in seeing your work!

[close the 2nd terminal tab by Xing it out or by running exit]

Move into the deploy/nginx folder

```
cd ~/Projects/MobyDock/deploy/nginx
```

Add the git remote for the production server

```
git remote add production ssh://you@x.x.x.x:/var/git/nginx.git
<Replace you with your username, or omit this if it's the same>
<Replace x.x.x.x with your droplet's IP address>
```

Push the code to the production server

```
git push production master
```

Move into the mobydock application folder

```
cd ~/Projects/MobyDock/mobydock
```

Add the git remote for the production server

```
git remote add production ssh://you@x.x.x.x:/var/git/mobydock.git
<Replace you with your username, or omit this if it's the same>
<Replace x.x.x.x with your droplet's IP address>
```

Push the code to the production server

```
git push production master
```

Move into the deploy folder

```
cd ~/Projects/MobyDock/deploy
```

Run the application on the production server

```
APP_ENV="production" SERVER_IP="x.x.x.x" SSH_USER="root"
KEY_USER="$(whoami)" ./deploy.sh -r
<Replace x.x.x.x with your droplet's IP address>
```

```
[open a 2nd terminal tab by pressing CTRL+SHIFT+T]
```

SSH into the production server

```
ssh you@x.x.x.x
<Replace you with your username, or omit this if it's the same>
<Replace x.x.x.x with your droplet's IP address>
```

Restart Docker once more

If you decide not to refactor the deploy.sh script, you could slightly alter it by automatically rebooting the box at the end of -a, this way you're never afflicted by the Docker restart issue on a fresh provision and it's a good idea to make sure your web application survives a reboot anyways

```
sudo systemctl restart docker
```

Confirm the newly running Docker containers

```
docker ps
```

Create the PostgreSQL database

```
docker exec postgres createdb -U postgres mobydock
```

^ If you see that it already exists, that's ok.

Create the database user and grant permissions onto the database

```
docker exec postgres psql -U postgres -c "CREATE USER mobydock WITH PASSWORD 'nocturnalpupilbacteriaexplosion'; GRANT ALL PRIVILEGES ON DATABASE mobydock to mobydock;"
```

^ If you see that it already exists, that's ok.

^ If you changed the database name, username or password in the `config/settings.py` file you will need to adjust the files in both this command as well as the `docker-compose.yml` file.

`[close the 2nd terminal tab by Xing it out or by running exit]`

Visit the /seed route in the xubuntu's web browser

https://x.x.x.x/seed

^ Change the IP address to match your droplet's IP address.

^ If you see a "This connection is Untrusted" warning, that's good! Click through to understand the risks.

`[follow the video to make an edit to the layout template]`

Make sure you're in the mobydock project folder

```
cd ~/Projects/MobyDock/mobydock
```

Update the git repo

```
git add -A && git commit -m "No longer shouting feed count"
```

Push the code change to the staging server

```
git push production master
```

Visit the / route in the xubuntu's web browser to view the changes

https://x.x.x.x/seed

^ Change the IP address to match your droplet's IP address.