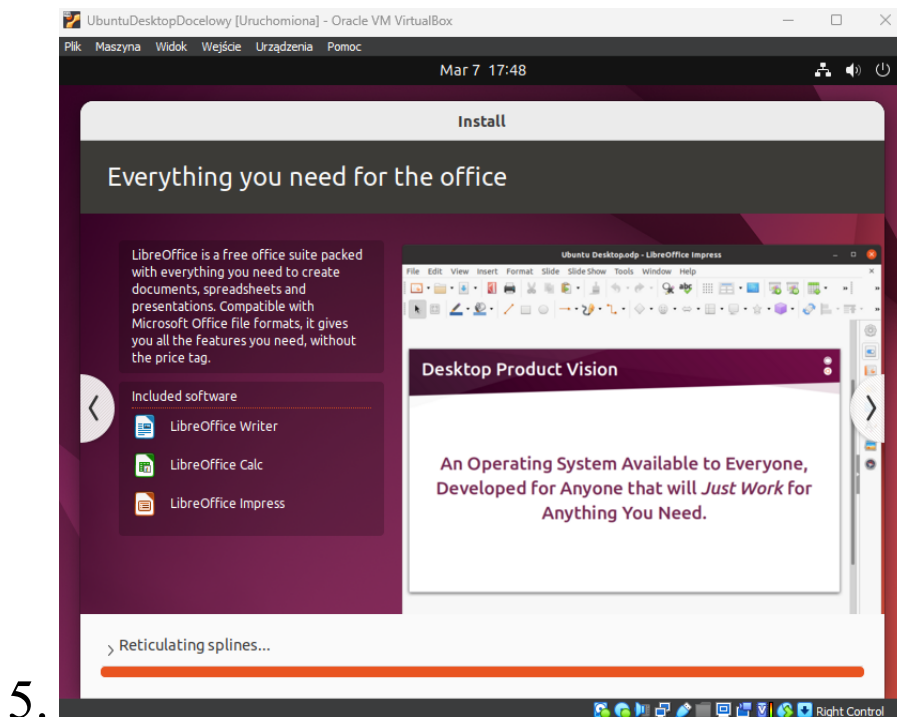


Challenge

“Postaw aplikacje flaskowa na Ubuntu, przy uzyciu dockera, uzyj obrazu Linux Alpine

1. Poczytałem o UBUNTU i systemach linuxowych,
2. Ściągnąłem VirtualBoxa i poczytałem jak postawić UBUNTU,
3. Po obejrzeniu tutorialu postawiłem UBUNTU, natomiast okazało się, że była to wersja serwerowa, więc zacząłem szukać wersji z GUI
4. Zainstalowałem już odpowiednią wersję,



6. Po uruchomieniu ubuntu, okazało się, że nie mogę włączyć terminala,
7. Po googlowaniu i sprawdzaniu różnych opcji, pomogło to:

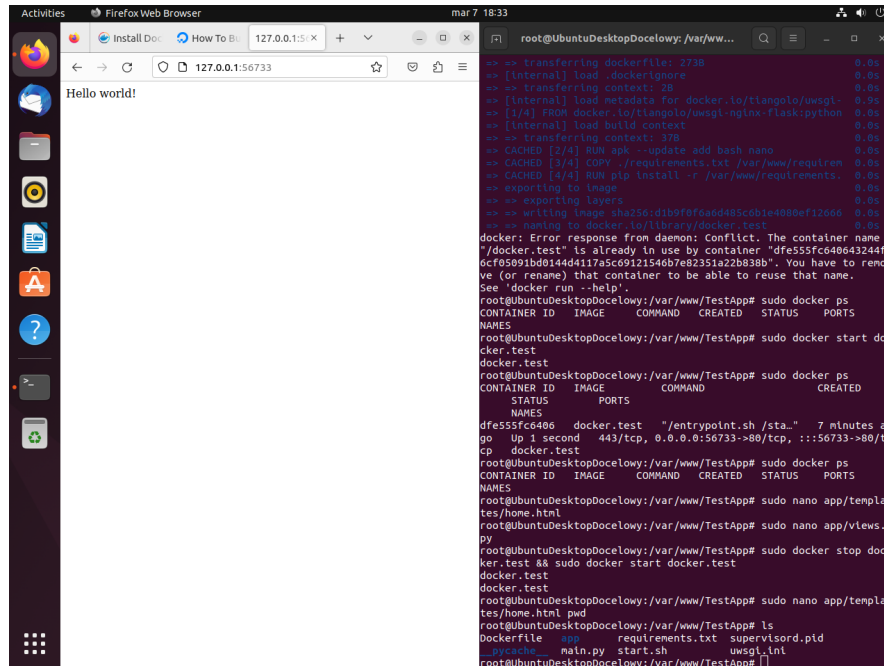


8. Już w działającym terminalu, zainstalowałem flaskę,
9. Zacząłem szukać informacji o dockerze i zainstalowałem docker engine korzystając z dokumentacji
<https://docs.docker.com/engine/install/ubuntu/>
10. Na koniec postawiłem całą aplikację dzięki temu

11. <https://www.digitalocean.com/community/tutorials/how-to-build-and-deploy-a-flask-application-using-docker-on-ubuntu-20-04>

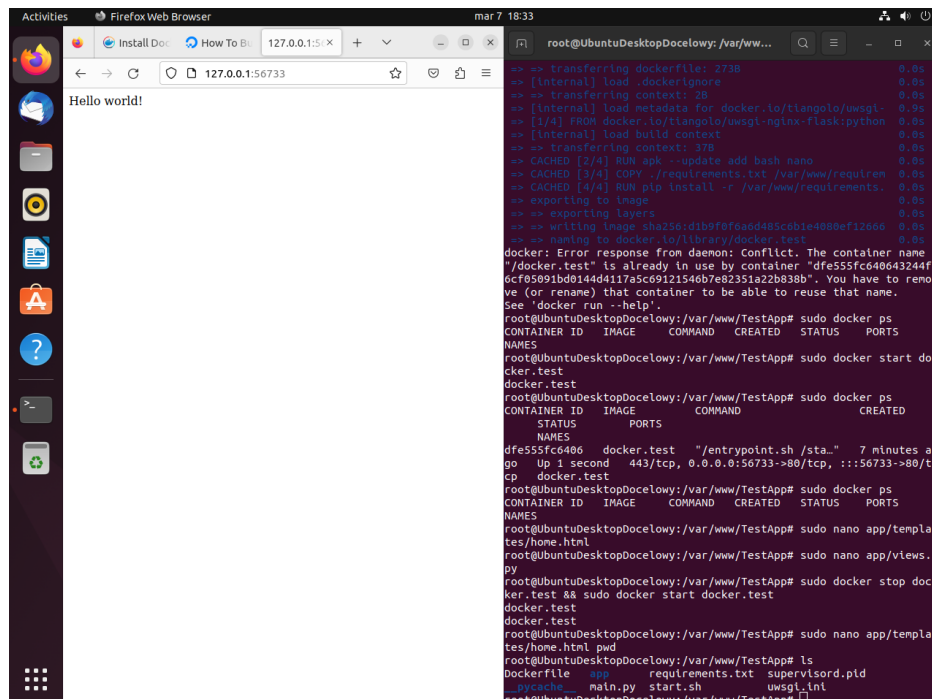
Efekt Koncowy

12.



```
root@UbuntuDesktopDoeLowy: /var/www...
==> transferring dockerfile: 273b
==> [internal] load .dockerignore
==> transferring context: 2B
==> [internal] load metadata for docker.io/tiangolo/uwsgi-
==> [3/4] FROM docker.io/tiangolo/uwsgi-nginx-flask-python
==> [internal] load build context
==> transferring context: 37B
==> CACHED [2/4] RUN apk --update add bash nano
==> CACHED [3/4] COPY ./requirements.txt /var/www/requirem
==> CACHED [4/4] RUN pip install -r /var/www/requirements.
==> exporting to image
==> exporting layers
==> writing image sha256:d1b9f06a6d485c0b1e4080ef12666
==> naming to docker.io/library/docker.test
docker: Error response from daemon: Conflict. The container name
"/docker.test" is already in use by container "dfe555fc640643244f
6cf05091bd0144d4117a5c69121546b7e82351a22b838b". You have to remo
ve (or rename) that container to be able to reuse that name.
See 'docker run --help'.
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS    PORTS
NAME
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker start do
cker.test
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS    PORTS
NAME
dfe555fc6406   docker.test "/entrypoint.sh /sta..." 7 minutes a
go Up 1 second 443/tcp, 0.0.0.0:56733->80/tcp, :::56733->80/t
cp docker.test
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS    PORTS
NAME
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo nano app/templa
tes/home.html
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo nano app/views.
py
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker stop doc
ker.test && sudo docker start docker.test
docker.test
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo nano app/templa
tes/home.html.pd
root@UbuntuDesktopDoeLowy: /var/www/TestApp# ls
Dockerfile  app  requirements.txt  supervisord.pid
__pycache__ main.py  start.sh          uwsgi.ini
root@UbuntuDesktopDoeLowy: /var/www/TestApp#
```

13.



```
root@UbuntuDesktopDoeLowy: /var/www...
==> transferring dockerfile: 273b
==> [internal] load .dockerignore
==> transferring context: 2B
==> [internal] load metadata for docker.io/tiangolo/uwsgi-
==> [3/4] FROM docker.io/tiangolo/uwsgi-nginx-flask-python
==> [internal] load build context
==> transferring context: 37B
==> CACHED [2/4] RUN apk --update add bash nano
==> CACHED [3/4] COPY ./requirements.txt /var/www/requirem
==> CACHED [4/4] RUN pip install -r /var/www/requirements.
==> exporting to image
==> exporting layers
==> writing image sha256:d1b9f06a6d485c0b1e4080ef12666
==> naming to docker.io/library/docker.test
docker: Error response from daemon: Conflict. The container name
"/docker.test" is already in use by container "dfe555fc640643244f
6cf05091bd0144d4117a5c69121546b7e82351a22b838b". You have to reno
ve (or rename) that container to be able to reuse that name.
See 'docker run --help'.
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS    PORTS
NAME
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker start do
cker.test
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS    PORTS
NAME
dfe555fc6406   docker.test "/entrypoint.sh /sta..." 7 minutes a
go Up 1 second 443/tcp, 0.0.0.0:56733->80/tcp, :::56733->80/t
cp docker.test
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  STATUS    PORTS
NAME
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo nano app/templa
tes/home.html
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo nano app/views.
py
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo docker stop doc
ker.test && sudo docker start docker.test
docker.test
root@UbuntuDesktopDoeLowy: /var/www/TestApp# sudo nano app/templa
tes/home.html.pd
root@UbuntuDesktopDoeLowy: /var/www/TestApp# ls
Dockerfile  app  requirements.txt  supervisord.pid
__pycache__ main.py  start.sh          uwsgi.ini
root@UbuntuDesktopDoeLowy: /var/www/TestApp#
```

14. Natomiast sprawdzając czy to jest na pythonie 3.8, mój przyjaciel Damian, zasugerował mi żebym usunął pythona z ubuntu, i spróbował zainstalować wersję 3.8. 😊 Nie wiedziałem jeszcze, że usunięcie skutkuje posypaniem się całego GUI, bo jak się okazało python jest jego dość ważnym aspektem działania. Także po próbach reanimacji systemu, usunąłem serwer i postawiłem jeszcze raz i aplikacje powtórzyłem również. 😊