**Load Balancing NGINX**

Reference: <https://towardsdatascience.com/sample-load-balancing-solution-with-docker-and-nginx-cf1ffc60e644>

1. Spin up one AWS EC2 instances (probably small) with Ubuntu installed.
2. Create three folders, app1, app2, and nginx, as well as a file docker-compose.yml. Leave this blank for now
3. Inside app1 add the following with the code specified underneath
   1. Dockerfile

FROM python:3

COPY ./requirements.txt /requirements.txt

WORKDIR /

RUN pip install -r requirements.txt

COPY . /

ENTRYPOINT [ "python3" ]

CMD [ "app1.py" ]

* 1. app1.py

from flask import request, Flask

import json

app1 = Flask(\_\_name\_\_)

@app1.route('/')

def hello\_world():

return 'Salam alikom, this is App1 :) '

if \_\_name\_\_ == '\_\_main\_\_':

app1.run(debug=True, host='0.0.0.0')

* 1. requirements.txt

Flask==1.1.1

1. Inside app2 add the following with the code specified underneath
   1. Dockerfile

FROM python:3

COPY ./requirements.txt /requirements.txt

WORKDIR /

RUN pip install -r requirements.txt

COPY . /

ENTRYPOINT [ "python3" ]

CMD [ "app1.py" ]

* 1. app2.py

from flask import request, Flask

import json

app1 = Flask(\_\_name\_\_)

@app1.route('/')

def hello\_world():

return 'Salam alikom, this is App2 :) '

if \_\_name\_\_ == '\_\_main\_\_':

app1.run(debug=True, host='0.0.0.0')

* 1. Requirements.txt

Flask==1.1.1

1. Inside nginx add the following with the code specified underneath
   1. Dockerfile

FROM nginx

RUN rm /etc/nginx/conf.d/default.conf

COPY nginx.conf /etc/nginx/conf.d/default.conf

* 1. Nginx.conf

upstream loadbalancer {

server 172.17.0.1:5001 weight=6;

server 172.17.0.1:5002 weight=4;

}

server {

location / {

proxy\_pass [http://loadbalancer;](about:blank)

}}

1. Edit docker-compose.yml to the following

version: '3'

services:

app1:

build: ./app1

ports:

- "5001:5000"

app2:

build: ./app2

ports:

- "5002:5000"

nginx:

build: ./nginx

ports:

- "8080:80"

depends\_on:

- app1

- app2

1. Spin the whole thing up with docker-compose up
2. Access the instance from port 8080