Nginx Load Balancer

load balancer is like a traffic controller for servers, making sure that the workload is evenly distributed and everyone gets served quickly and efficiently.

for setting up a load balancer with NGINX webserver (NGINX docker image) :

1- make 3 different html files with different background and name them index.html.(green page in this example)

```
DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: green;
}
</style>
</head>
<body>

<h1>This is a sample HTML page with a green background</h1>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque nec hendrerit mi. Integer euismod commodo ni

</body>
</html>

```

```
mkdir html
cd html
mkdir green blue red
nano index.html
cp index.html green/ #also for other directories
```

2- Now make 3 containers for serving your html files

- in this case you can use docker compose and nginx image to serve html files.

  ```
  nano docker-compose.yaml
  ```

- add these lines to your docker-compose.yaml file.

  ```
  version: '3'
  services:
    load_balancer:
      container_name: lb
      image: nginx
      ports:
        - 80:80
      volumes:
        - ./nginx.conf:/etc/nginx/nginx.conf:ro
      restart: always
      networks:
        webnet:
          ipv4_address: 192.168.1.3
    #web 1 green
    web1:
      image: nginx
  ```

```
    container_name: web1
    restart: always
    volumes:
      - ./html/green:/usr/share/nginx/html
    networks:
      webnet:
        ipv4_address: 192.168.1.7
  #web 2 blue
  web2:
    image: nginx
    container_name: web2
    restart: always
    volumes:
      - ./html/blue:/usr/share/nginx/html
    networks:
      webnet:
        ipv4_address: 192.168.1.8
  #web 3 red
  web3:
    image: nginx
    container_name: web3
    restart: always
    volumes:
      - ./html/red:/usr/share/nginx/html
    networks:
      webnet:
        ipv4_address: 192.168.1.9

networks:
  webnet:
    ipam:
      config:
        - subnet: 192.168.1.0/24
```

- add network for all containers and give them a static ip.

3- Make a nginx.conf file and write load balancing rules in the file.

```
nano nginx.conf
```

- nginx.conf file should be like this:

```
http {
  upstream backend {
    server 192.168.1.7 weight=20;
    server 192.168.1.8 weight=10;
    server 192.168.1.9 weight=1;
  }

  server {
    listen 80;

    location / {
      proxy_pass http://backend;
      proxy_set_header Host $host;
     }
    }
}
   events {
       worker_connections 1024;
```

```
19    }
20
```

4- type this command to run your servers and load balancer:

```
1  docker-compose -f docekr-compose.yaml up -d
```