

Beskrivelse:

Dette er kun en lille demo. Indtil nu har vi jo kun haft nemme containere. Kan man egentlig også have lidt mere avancerede funktioner i de der containere noget? Ja selvfølgelig kan man det. Det er jo ikke containeren, eller imaget, der sætter begrænsningen. Det er hvad man kan lave med det der er i vores containere og images. Denne lille demonstration vil vise en simpel opsætning af load balancing ved hjælp af en nginx proxy. Normalt vil man selvfølgelig ikke kun have vores web containere liggende på én fysisk server, men dele dem mellem flere fysiske servere ved hjælp af et cluster, eller som det hedder i docker terminologi: En dockwer Swarm! Cluster opsætning vil jeg ikke gennemgå denne gang. Her demonstrer jeg det med en proxy og 5 webservere bag ved, hvor belastningen bliver delt ud mellem alle webservere.

Lad os komme i gang:

Log på din worker, efter den er nulstillet.

Først opretter vi et bibliotek og går ned i det:

```
mkdir loadbalancing  
cd loadbalancing
```

Eksempel output:

```
dtmek@docker2:~$ mkdir loadbalancing  
cd loadbalancing  
dtmek@docker2:~/loadbalancing$
```

Så kopierer vi alle filer over til vores docker workstation fra opgave 6 filer, og selvfølgelig ned i underbiblioteket loadbalanceing, og ser at vi har fået dem lagt over:

```
ls -l
```

Eksempel output:

```
dtmek@docker2:~/loadbalancing$ ls -l  
total 32  
-rw-rw-r-- 1 dtmek dtmek 511 Nov 2 20:52 default.conf  
-rw-rw-r-- 1 dtmek dtmek 1587 Nov 18 11:44 docker-compose2.yml  
-rw-rw-r-- 1 dtmek dtmek 1587 Nov 18 11:22 docker-compose.yml  
drwxrwxr-x 2 dtmek dtmek 4096 Nov 20 10:13 web1  
drwxrwxr-x 2 dtmek dtmek 4096 Nov 20 10:13 web2  
drwxrwxr-x 2 dtmek dtmek 4096 Nov 20 10:13 web3  
drwxrwxr-x 2 dtmek dtmek 4096 Nov 20 10:13 web4  
drwxrwxr-x 2 dtmek dtmek 4096 Nov 20 10:13 web5  
dtmek@docker2:~/loadbalancing$
```

Vi kan se at der også ligger en docker-compose.yml fil. Hvis man kigger på filen (**nano docker-compose.yml**) kan man se at alle vores web servere peget til den samme webside. Det er jo også det vi gerne vil have. Så der er flere webservere der deles om arbejdet.

Lad os kortvarigt kigge på default.conf. Det er måden man opsætter loadballancing. (Ingen eksempel) Man kan altid finde mere information ved at søge på upstream nginx på google. Her kan man bl.a. finde ud af hvordan man forbedrer systemet ved f.eks. at gå over til en helt anden webside, hvis de normale ikke kan kontaktes. F.eks. en side der siger: "Ingen kontakt, prøv senere" Nok beskrivelse for denne gang, lad os starte det hele:

```
docker compose up -d
```

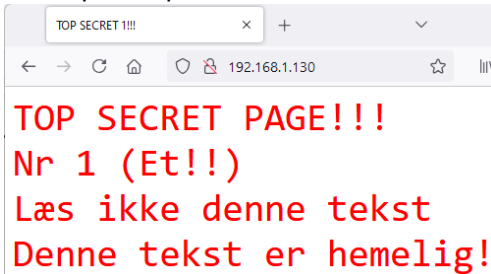
Eksempel output:

```
dtmek@docker2:~/loadbalancing$ docker compose up -d
[+] Running 13/13
  ✔ proxy1 Pulled
  ✔ web2 Pulled
  ✔ web4 Pulled
  ✔ web3 Pulled
  ✔ web5 Pulled
  ✔ 2d429b9e73a6 Pull complete
  ✔ 9b1039c85176 Pull complete
  ✔ 9ad567d3b8a2 Pull complete
  ✔ 773c63cd62e4 Pull complete
  ✔ 1d2712910bdf Pull complete
  ✔ 4b0adc47c460 Pull complete
  ✔ 171eebbdf235 Pull complete
  ✔ web1 Pulled
[+] Running 7/7
  ✔ Network web Created
  ✔ Container web2 Started
  ✔ Container proxy1 Started
  ✔ Container web3 Started
  ✔ Container web4 Started
  ✔ Container web1 Started
  ✔ Container web5 Started
dtmek@docker2:~/loadbalancing$
```

Det startede jo fint. Lad os prøve at besøge vores hjemmeside:

```
http://ip_på_din_docker_maskine
```

Eksempel output:



Selv hvis du trykker F5 et par gange er det jo samme side. Hvordan kan vi så se om belastningen er fordelt over alle vores webservere? Jo vi bruger vores docker stats kommando:

```
docker stats
```

Eksempel output:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
cf3ba4232949	web3	0.00%	4.527MiB / 7.757GiB	0.06%	35.7kB / 20.4kB	0B / 12.3kB	5
b956a7f6ea26	web2	0.00%	4.555MiB / 7.757GiB	0.06%	36.9kB / 19.5kB	0B / 12.3kB	5
fc75d999c891	web4	0.00%	4.527MiB / 7.757GiB	0.06%	35.5kB / 18.8kB	0B / 12.3kB	5
f5456ae8d5c1	web1	0.00%	4.535MiB / 7.757GiB	0.06%	36.8kB / 21.5kB	0B / 12.3kB	5
3aecf60ce329	web5	0.00%	4.559MiB / 7.757GiB	0.06%	32kB / 16.8kB	0B / 12.3kB	5
d939267a456a	proxy1	0.00%	4.582MiB / 7.757GiB	0.06%	236kB / 233kB	0B / 4.1kB	5

Men... Der sker jo ikke noget? Næææ... vi laver jo heller ikke noget. Placer nu din webbrowser, så du kan se websiden, og samtidig se belastningen på maskinerne. Samtidig med at trykke på **F5** eller **CTRL + F5** i webvinduet, og holder tasterne nede:

Eksempel output:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
aed949ef4f48	web2	0.33%	4.477MiB / 7.757GiB	0.06%	196kB / 120kB	0B / 4.1kB	5
2e1fa4473cef	proxy1	3.27%	4.66MiB / 7.757GiB	0.06%	1.41MB / 1.4MB	0B / 4.1kB	5
1e47700f0d36	web5	0.31%	4.5MiB / 7.757GiB	0.06%	192kB / 120kB	0B / 4.1kB	5
ace107856051	web3	0.35%	4.496MiB / 7.757GiB	0.06%	195kB / 125kB	0B / 4.1kB	5
24b9d9db0e3	web1	0.32%	4.512MiB / 7.757GiB	0.06%	196kB / 125kB	0B / 4.1kB	5
3ce7a2c703ef	web4	0.37%	4.477MiB / 7.757GiB	0.06%	192kB / 120kB	0B / 4.1kB	5



Belastningsgraden vil selvfølgelig være afhængig af hvilke ressourcer din maskine har til rådighed. Men man kan se at det ca. er samme belastning for hver af web serverne. Proxy server er selvfølgelig den mest belastede, da den behandler trafikken til alle webservere.

Men vi kan gøre det tydeligere. Tilbage til vores SSH, og tryk på **CTRL + c** for at afbryde, og så vi stopper lige alt igen:

docker compose down

Eksempel output:

```
dtmek@docker2:~/loadbalancing$ docker compose down
[+] Running 7/7
  ✓ Container web5      Removed
  ✓ Container web1      Removed
  ✓ Container web4      Removed
  ✓ Container proxy1    Removed
  ✓ Container web2      Removed
  ✓ Container web3      Removed
  ✓ Network web         Removed
dtmek@docker2:~/loadbalancing$
```

Så skal vi lige have den anden docker-compose.yml fil klar. Det er en fil der sørger for at der er forskellige websider på hver af vores webservere. Så vi omdøber lige de to filer:

```
mv docker-compose.yml docker-composeold.yml
mv docker-compose2.yml docker-compose.yml
```

Eksempel output:

```
dtmek@docker2:~/loadbalancing$ mv docker-compose.yml docker-composeold.yml
dtmek@docker2:~/loadbalancing$ mv docker-compose2.yml docker-compose.yml
dtmek@docker2:~/loadbalancing$
```

Lad os starte det hele igen:

```
docker compose up -d
```

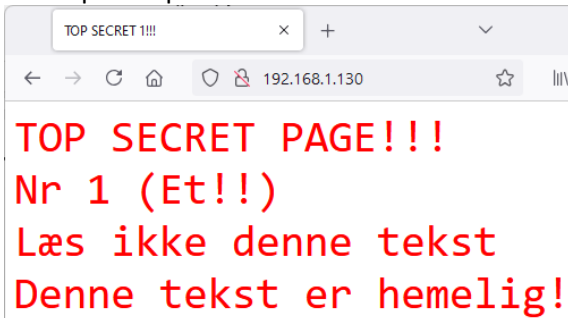
Eksempel output:

```
dtmek@docker2:~/loadbalancing$ docker compose up -d
[+] Running 7/7
  ✓ Network web          Created
  ✓ Container web3       Started
  ✓ Container web5       Started
  ✓ Container web1       Started
  ✓ Container web4       Started
  ✓ Container web2       Started
  ✓ Container proxy1     Started
dtmek@docker2:~/loadbalancing$
```

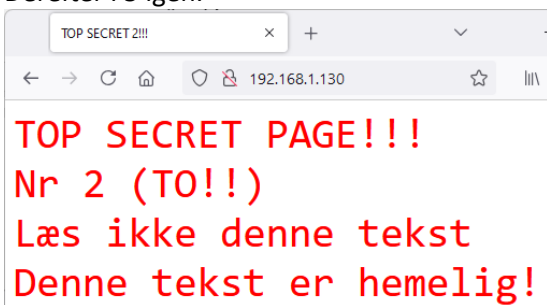
Nu burde der jo vises en anden side hver gang dy trykker på F5. lad os prøve at besøge vores webside(Den behøver ikke starte på nr.1):

```
http://ip_på_din_docker_maskine
```

Eksempel output:



Derefter F5 igen:



Osv.

Som man nu kan se er det faktisk en ny side der kommer frem hver gang man trykker F5, så det viser jo tydeligt at proxy server deler belastningen mellem de webservere der er til rådighed.

For at vende tilbage til det jeg skrev i indledningen omkring Kubernetes (K8s). det er så her at de containere vil være spredt ud over flere fysiske servere, og hvis belastningen på vores hjemmeside stiger, kan den starte flere webserver containere. Og f.eks. fordele dem ud over de fysiske servere der er til rådighed. Det er det der er styrken i Docker!

Dette var Opgave 6. Du er nu færdig med materialet omkring Docker.