

# AVISI

**USE DOCKER THEY SAID  
IT'LL BE FUN THEY SAID**

door: [Henk Jan Agteresch](#) / [@hj8rs](#)

# AVISI

## WIE BEN IK

- Henk-Jan Agteresch
- henkjan in #nlnoG
- Infrastructure Engineer bij Avisi
- Koninklijke Notariele beroepsorganisatie

# DOCKER IN HET KORT

- Het isoleren van processen op een machine (Docker container)
- Configuratie + applicatie + OS packages samen ge-deployed (Docker image)

**DOCKER WILL ALWAYS RUN  
THE SAME, REGARDLESS OF  
WHERE IT'S DEPLOYED**

~~DOCKER WILL ALWAYS RUN  
THE SAME, REGARDLESS OF  
WHERE IT'S DEPLOYED~~

---

# How to draw an Owl.

---

*"A fun and creative guide for beginners"*

---



Fig 1. Draw two circles

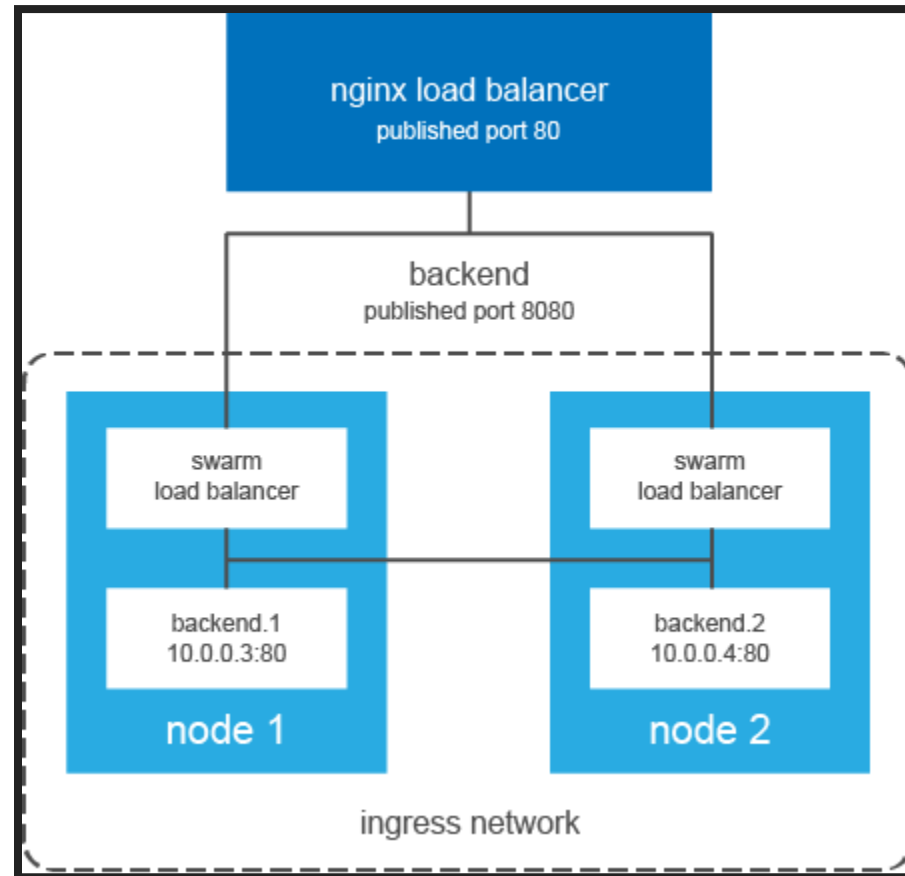


Fig 2. Draw the rest of the damn Owl

---

# **CLUSTERING**

# SWARM MODE





# UPGRADE REVERSE PROXYS

- Apache upgrade 2.2 -> 2.4
- SNI
- Uitrol met docker containers

# NO SPACE LEFT ON DEVICE #1

- disk usage onder de 20%
- geen quotas

# SEMAPHORES

- DOCKER OP OS-X

```
$ docker run --rm -it alpine sysctl kernel  
kernel.sem=32000 1024000000 500 32000
```

- DOCKER OP CENTOS-7

```
$ docker run --rm -it alpine sysctl kernel  
kernel.sem = 250 32000 32 128
```

# SEMAPHORES OPHOGEN

```
version: '2.1'
services:
  app:
    build: .
    sysctls:
      - kernel.sem=32000 1024000000 500 3
```

- Werkt niet onder swarm mode

# WAAROM ZOVEEL SEMAPHORES

- Default minimaal 8 semaphoren
- Elke gedefinieerde vhost = 1 semaphore
- Elke balancer gedefineerd in een vhost = 1 semaphore.
- $(\text{aantal balancers} * \text{aantal vhosts}) + (8 + \text{aantal vhosts})$

Opgelost door balancer in bijbehorende vhost te definiëren

# VERDERE ONFIGURATIE

```
- type: bind
  source: ./balancers
  target: /usr/local/apache2/balancers
  read_only: true
- type: bind
  source: ./files/brn-balancer.conf
  target: /usr/local/apache2/balancers/brn-balancer.conf
  read_only: true
```

Configuratie fout -> Docker daemon herstart, containers draaiden weer.

# NO SPACE LEFT ON DEVICE #2

2 dagen later

```
ERROR: for proxyst_test-proxy_1 Cannot start service test-proxy:  
oci runtime error: open /run/runc/79988ccede4e5e318bea226902b06407  
8fe70ec989b72248f0cf8eb2314de60b/state.json: no space left on device
```

# DF -H

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vg0-root 7.8G  2.4G  5.0G  32% /
devtmpfs        482M   0    482M   0% /dev
tmpfs           492M   0    492M   0% /dev/shm
tmpfs           492M  51M   442M  11% /run
tmpfs           492M   0    492M   0% /sys/fs/cgroup
/dev/xvda1      976M  152M  758M  17% /boot
```



# INODES

```
$ df -i
Filesystem          Inodes  IUsed  IFree  IUse% Mounted on
/dev/mapper/vg0-root 524288 105892 418396   21% /
devtmpfs            123235    366 122869    1% /dev
tmpfs                125848     1 125847    1% /dev/shm
tmpfs                125848 125847     1 100% /run
tmpfs                125848    16 125832    1% /sys/fs/cgroup
/dev/xvda1           65536    345 65191    1% /boot
```

# CRASHENDE HEALTHCHECK

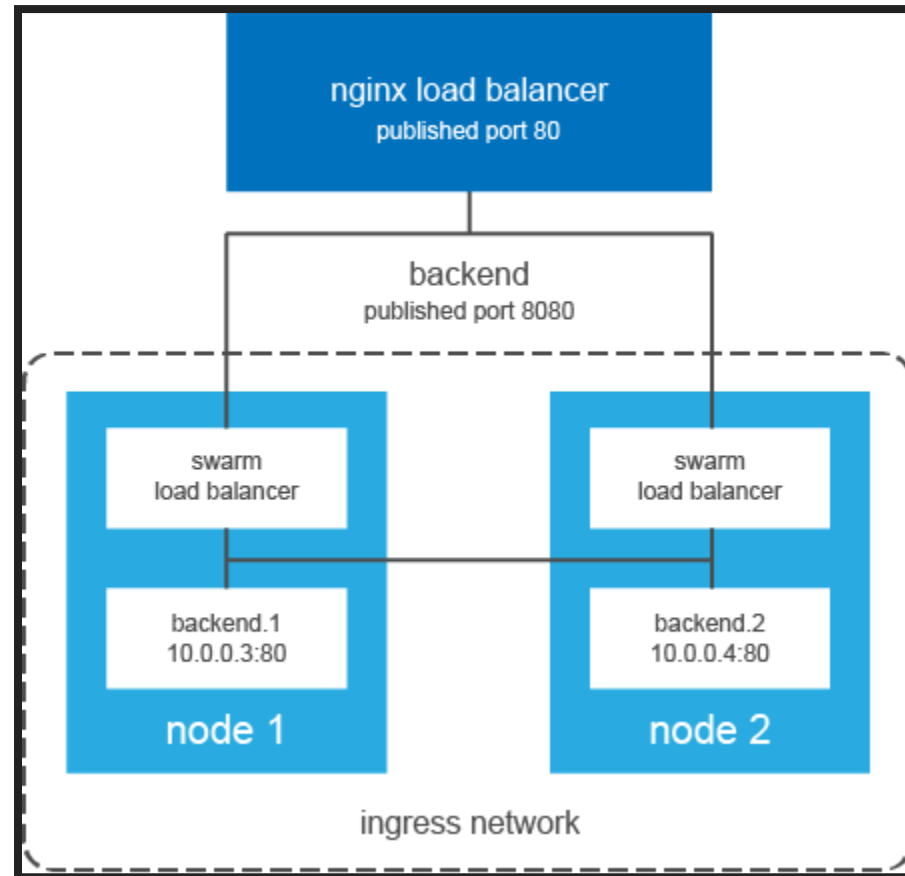
fifo /run/docker/libcontainerd/\$id/init-st{din,derr,out}

Veel fifo's -> veel inodes.

# SWARM MODE

- Ingress network
- Eenvoudige orchestration en deployment

# INGRESS NETWORK



# ORCHESTRATION EN DEPLOYMENT

```
$ docker service create \  
  --replicas 3 \  
  --name redis \  
  --update-delay 10s \  
  redis:3.0.6
```

# SOURCE IP

```
$ ip -4 a s |grep 172.17
  inet 172.17.0.1/16 scope global docker0
```

```
$ docker run -d --name testproxy -p 8080:80 nginx:alpine
9dcb35f94ea5d643b1b46b16bc28bd3740d9632ed6e4865f49fd505927b3ba58
$ curl localhost:8080
Thank you for using nginx.
```

```
$ docker logs testproxy
172.17.0.1 - - [16/Jun/2017:08:47:10 +0000] "GET / HTTP/1.1" 200 612
```

# SOURCE IP SWARM MODE

```
$ docker service create --name swarmproxy -p 9080:80 nginx:alpine
$ curl localhost:9080
Thank you for using nginx.
```

```
$ docker service logs swarmproxy
swarmproxy.1.rvzbiprw7o0f@newton | 10.255.0.3 - - [16/Jun/2017:08
```

```
$ docker network ls|grep ingress
vrq681gfvft9          ingress                overlay                swarm
$ docker network inspect vrq681gfvft9|grep 10.255.0
    "Subnet": "10.255.0.0/16",
    "Gateway": "10.255.0.1"
    "IPv4Address": "10.255.0.3/16",
```

# SOURCE IP HOST MODE

```
$ docker service create --name hostmodeproxy --publish  
mode=host,target=80,published=7080 nginx  
$ curl localhost:7080  
Thank you for using nginx.
```

```
$ docker service logs hostmodeproxy  
hostmodeproxy.1.jvd4wl5waxcb@newton | 172.17.0.1 - - [16/Jun/2017
```



# RESULTAAT SWARM + HOSTMODE

- Geen ingress netwerk
- Geen Swarm load-balancing van buiten af voor die poort
- Client remote IP wordt juist doorgegeven

**VRAGEN ?**