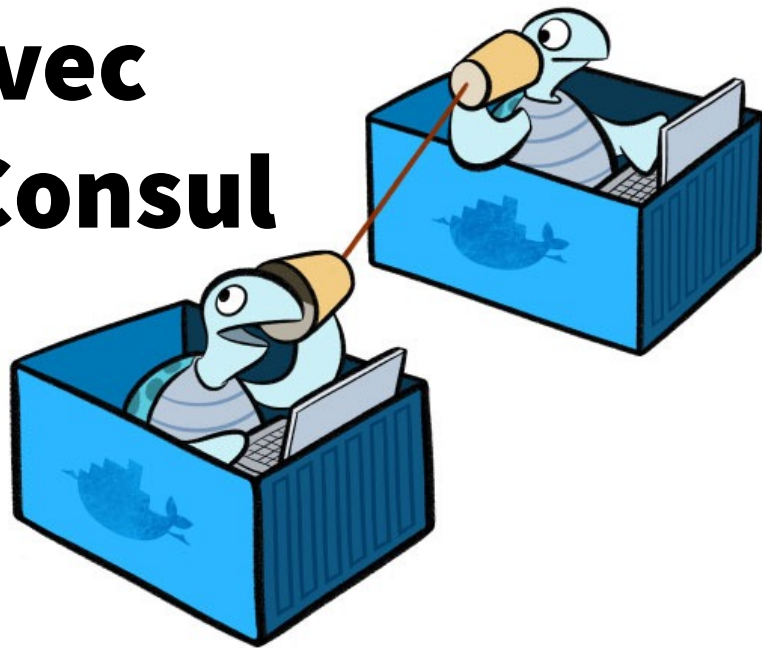
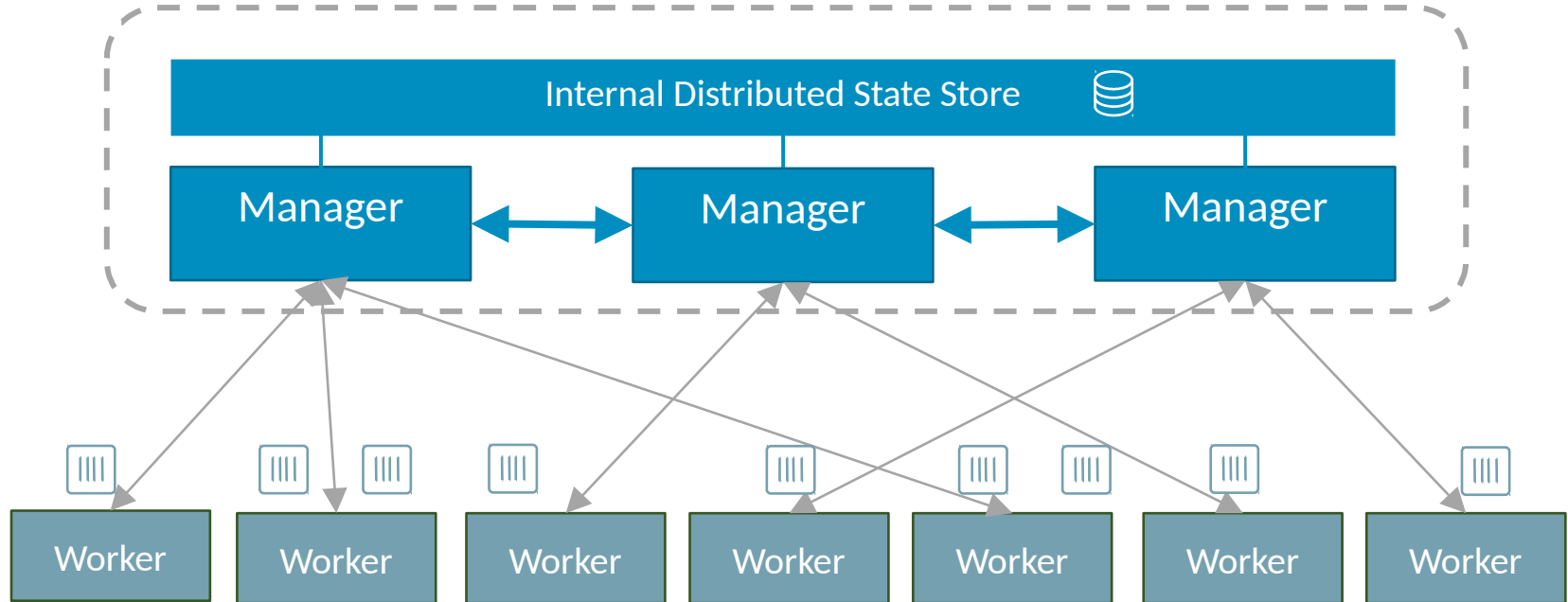


Load-balancing avec Swarm-mode et Consul

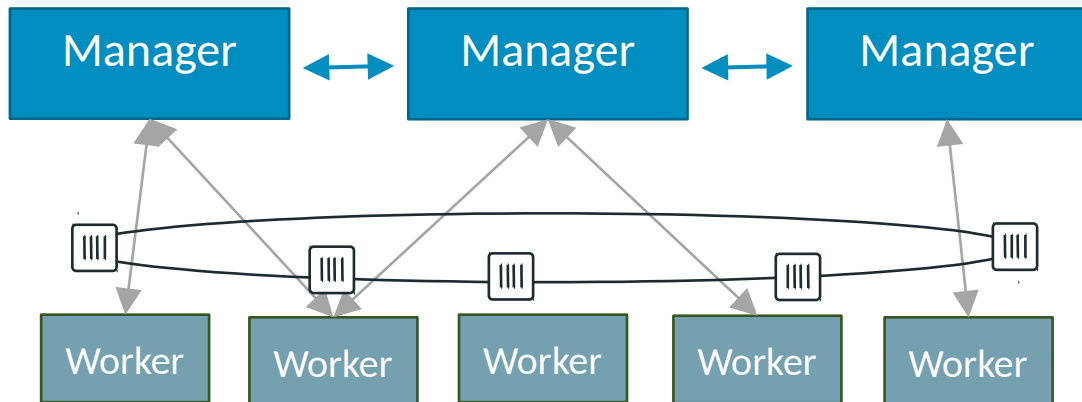
Alex Drahon – Solution Architect
alex.drahon@docker.com



Rappel: Swarm-mode



Swarm-mode services



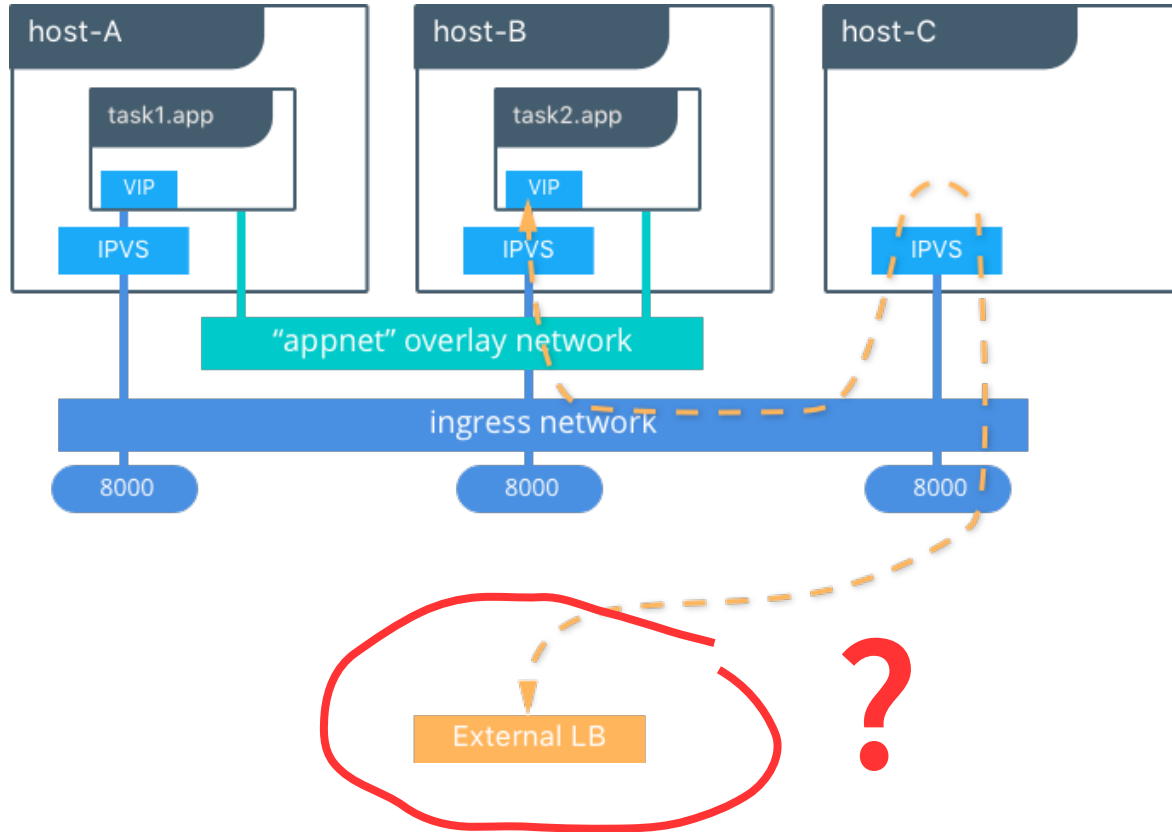
The declarative command describes a new service:

- Named nginx
- Made of 5 containers based on my nginx-rand image
- Connected on an overlay network called “myoverlay”
- Assigned to port 80

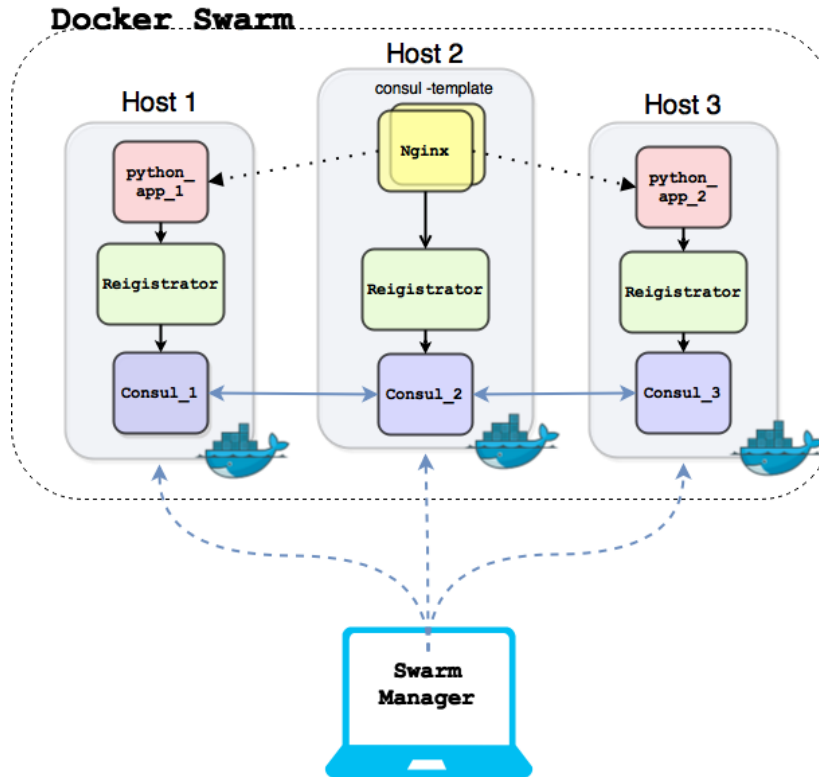
```
$ docker service create --replicas 5 --name nginx  
--network myoverlay --publish 80/tcp adrahon/nginx-  
rand
```



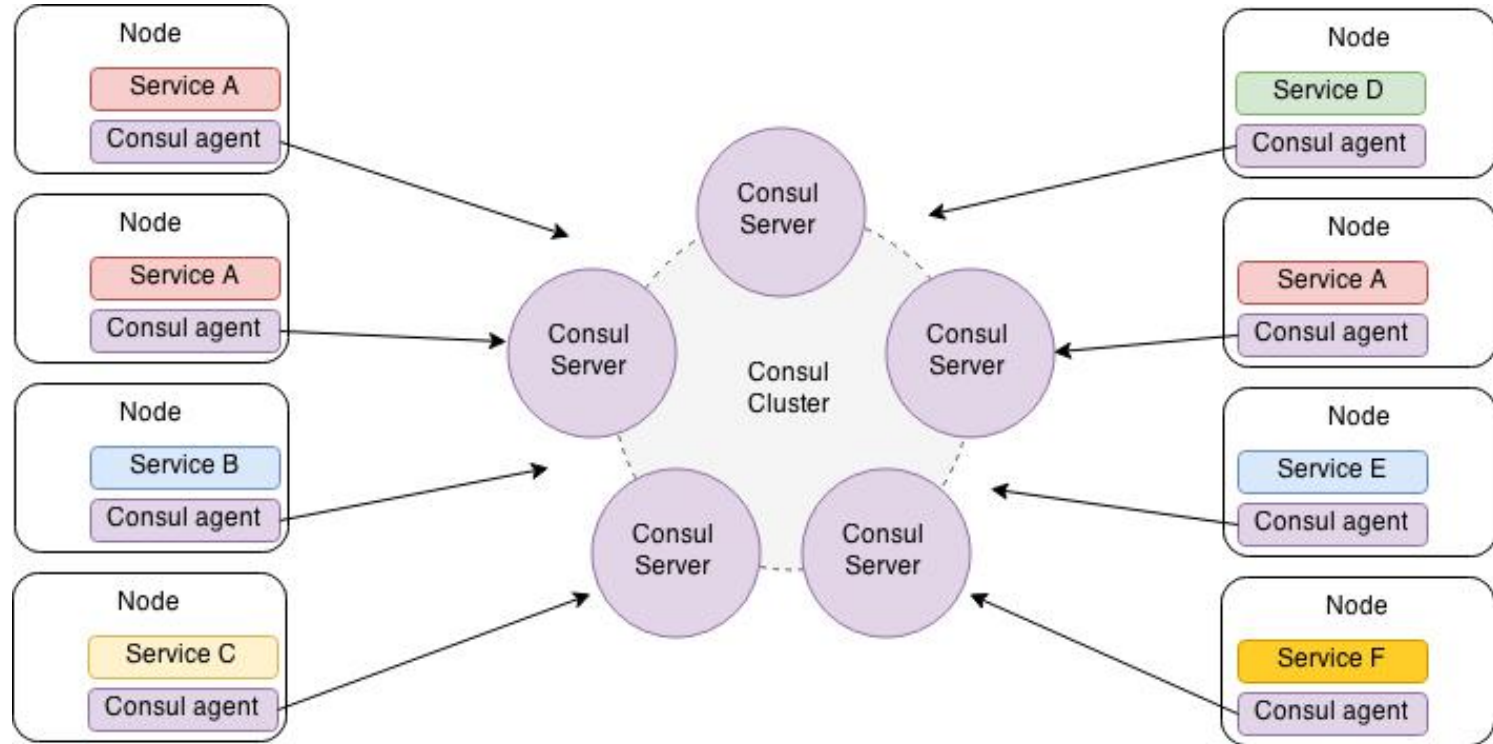
Overlay networking: routing mesh



L'ancien pattern



1er problème: Consul



Consul: le cluster

Cluster Consul dans son propre overlay, DNS RR pour la discovery

```
docker network create --driver overlay consul
```

```
docker service create -e CONSUL_BIND_INTERFACE='eth0' \  
  --name consul-cluster --network consul \  
  -endpoint-mode dnsrr --replicas 3 \  
  consul agent -server -client=0.0.0.0 \  
  -bootstrap-expect=3 -retry-join=consul-cluster
```



Consul: les agents

Agent exposé, 1 replica mais scalable

```
docker service create -e CONSUL_BIND_INTERFACE='eth0' \  
  --name consul-agent --network consul --replicas 1 \  
  -p 8500:8500 \  
  consul agent -client=0.0.0.0 -retry-join=consul-cluster
```



2e problème: Registrator

Registrator ne voit pas les services, seulement les containers

Les services ne génèrent pas d'évènements (pour l'instant)

Registrator, comme Consul, est host-based (service address)

```
HOST_IP=$(ip addr | grep 'state UP' -A2 | tail -n1 | \
    awk '{print $2}' | cut -f1 -d'/')
```

(en général)

```
docker run -d --name=registrator -net=host -restart=always \
    volume=/var/run/docker.sock:/tmp/docker.sock \
    gliderlabs/registrator:latest -ip $HOST_IP \
    consul://localhost:8500
```



Le load balancer

Haproxy + Consul template, nginx-upsync-module, Traefik, Fabio...

Avec Fabio:

```
echo "registry.consul.addr = mynode.example.com:8500" \  
> fabio.properties
```

```
docker run -d -p 9999:9999 -p 9998:9998 \  
-v $PWD/fabio.properties:/etc/fabio/fabio.properties \  
magiconair/fabio
```

Exercice: Fabio comme service



Créer un service

Pour Registrator:

- le service utilise le host-mapping pour être détecté
- on passe des paramètres avec `--container-label`

Pour Fabio:

- on crée un healthcheck
- on passe un paramètre `urlprefix`

```
docker service create --name web \  
  --container-label SERVICE_80_NAME=webs \  
  --container-label SERVICE_80_CHECK_HTTP=/ \  
  --container-label SERVICE_TAGS=urlprefix-/ \  
  --publish mode=host,published=5000,target=80,protocol=tcp \  
  --replicas 2 adrahon/nginx-rand
```





THANK YOU