



# Docker Swarm



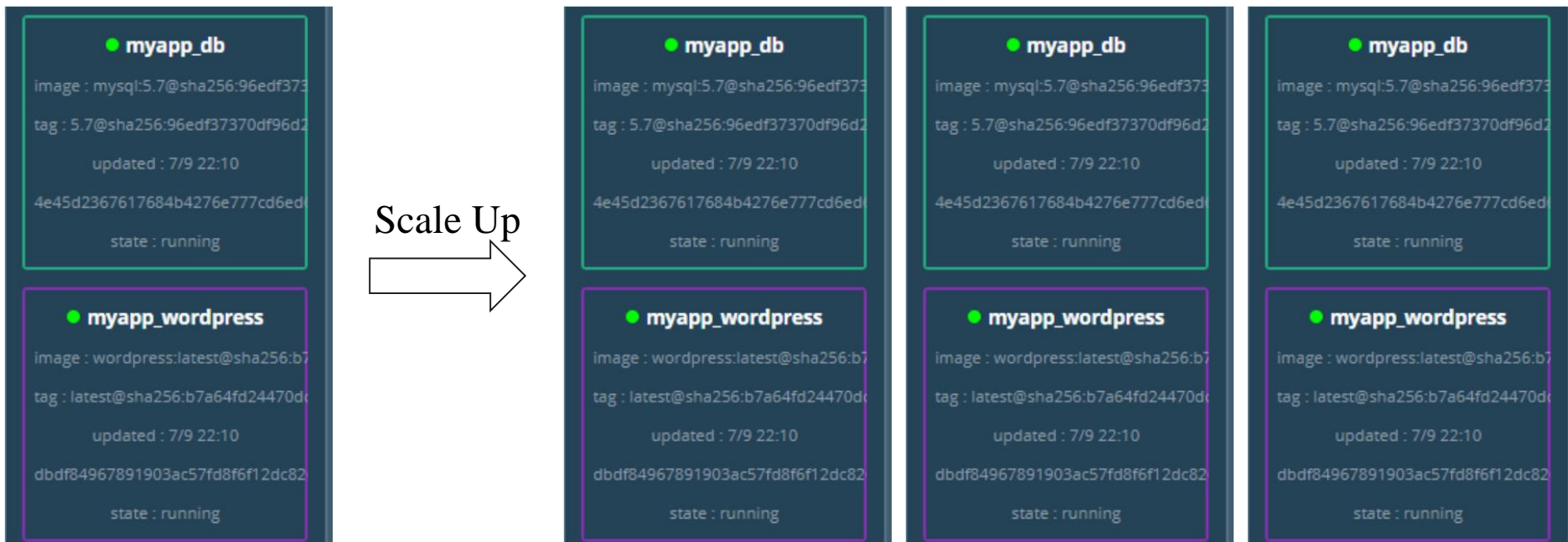
# Agenda

- **Swarm Mode**
- **Khởi tạo dịch vụ và mở rộng**
- **Khởi tạo 3 node-swarm cluster**
- **Mở rộng với Overlay Networking**





# Chạy N containers trên M hosts ?



Docker Host

Docker Hosts



# Chạy N containers trên M hosts ?



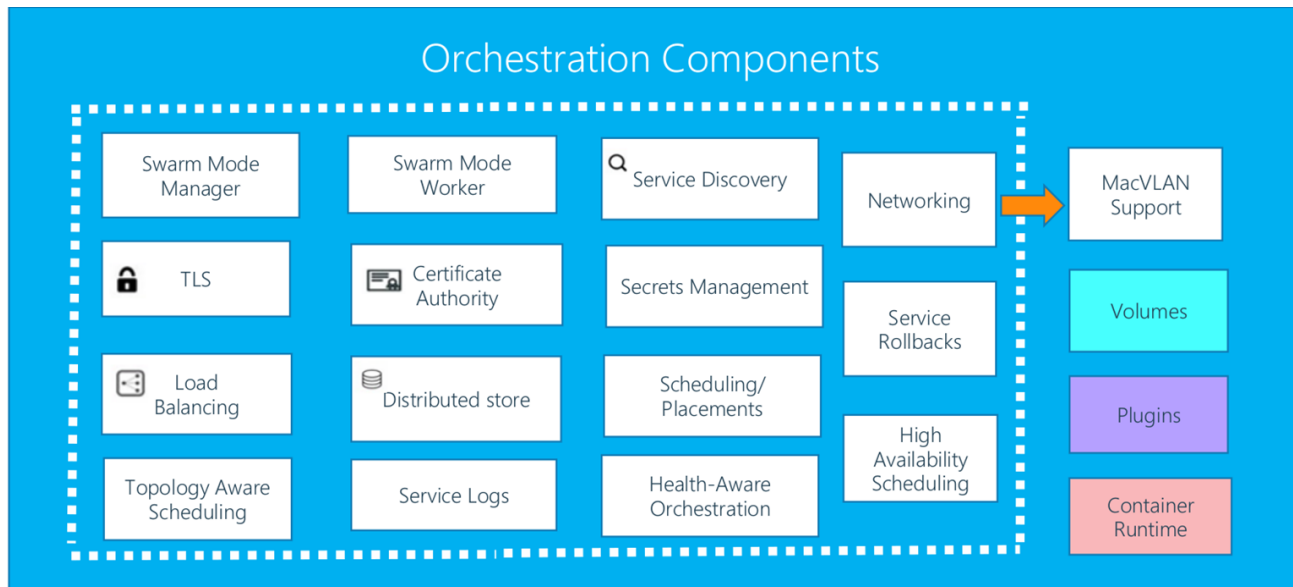
Docker Hosts



# Swarm Mode: Build-in orchestration

## Key concepts:

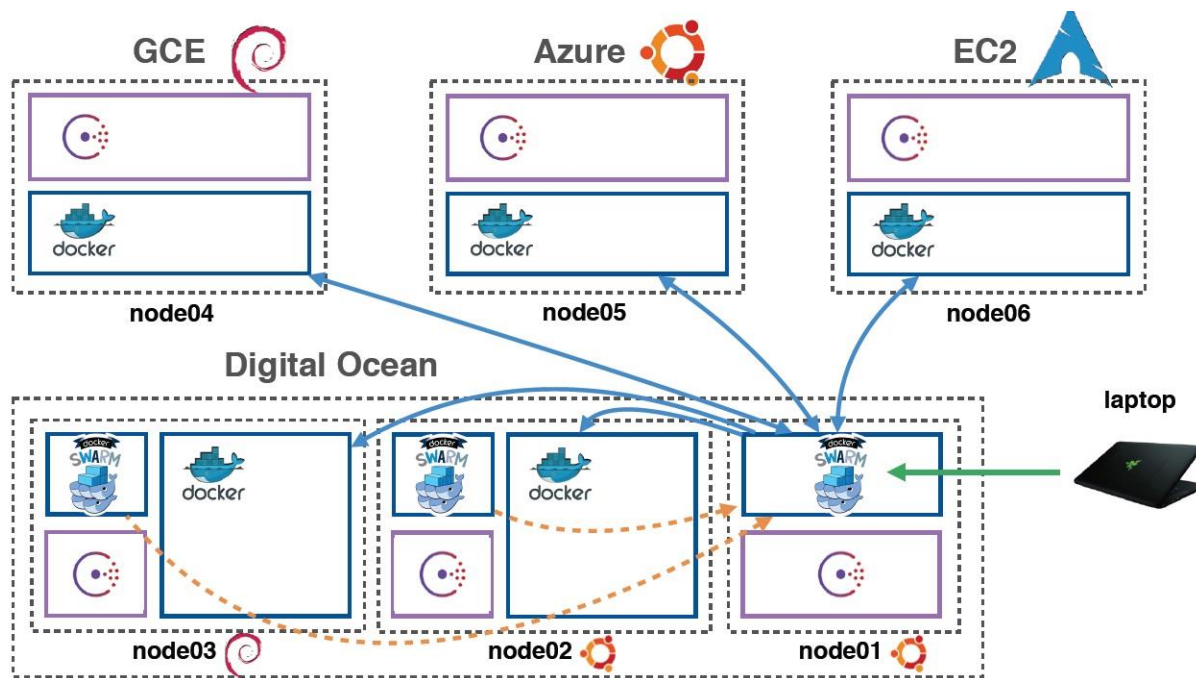
- Node:
  - Manager
  - Worker
- Service and task





# Swarm Mode: Build-in orchestration

**Nodes:** một Docker Engine instance tham gia vào Swarm



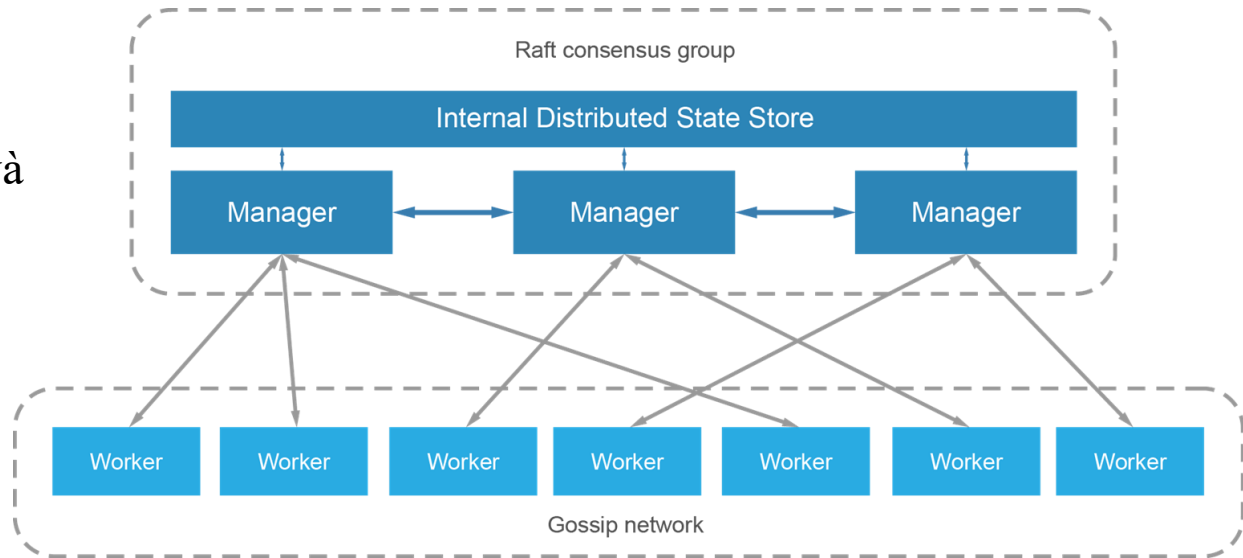


# Swarm Mode: Build-in orchestration

## Manager:

- + Thực hiện việc quản lý và điều phối cluster
- + Bầu một leader cho việc điều phối tasks

**Worker:** nhận và thực hiện tasks từ Manager node



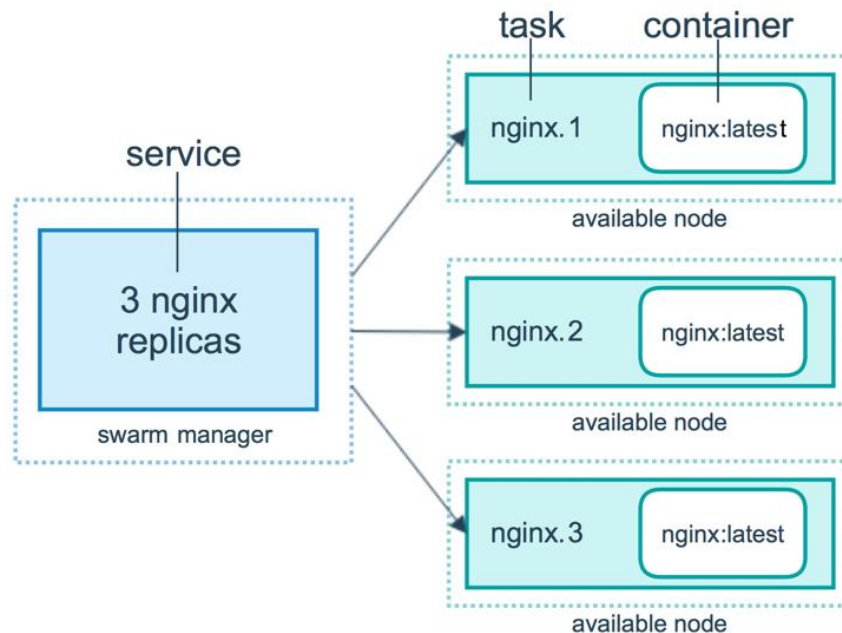
Mặc định Manager nodes cũng có thể chạy services như Worker nodes



# Swarm Mode: Build-in orchestration

Một **Services** chỉ định:

- Port sử dụng cho các dịch vụ ở ngoài Swarm
- Một Overlay network cho các dịch vụ kết nối với nhau trong Swarm
- Giới hạn tài nguyên CPU và memory
- Rolling update policy
- Số lượng các bản Replicas của Image trong Swarm



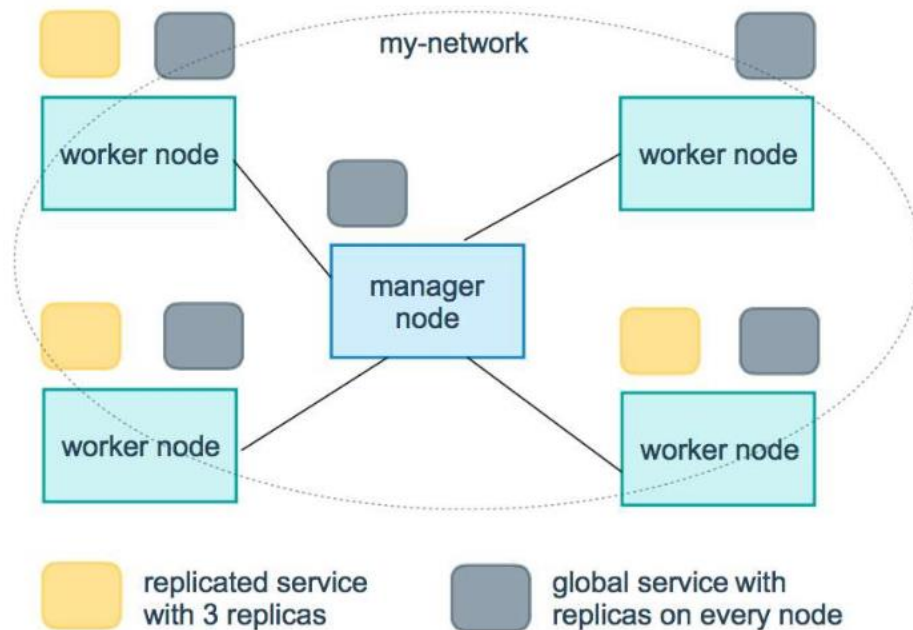




# Swarm Mode: Build-in orchestration

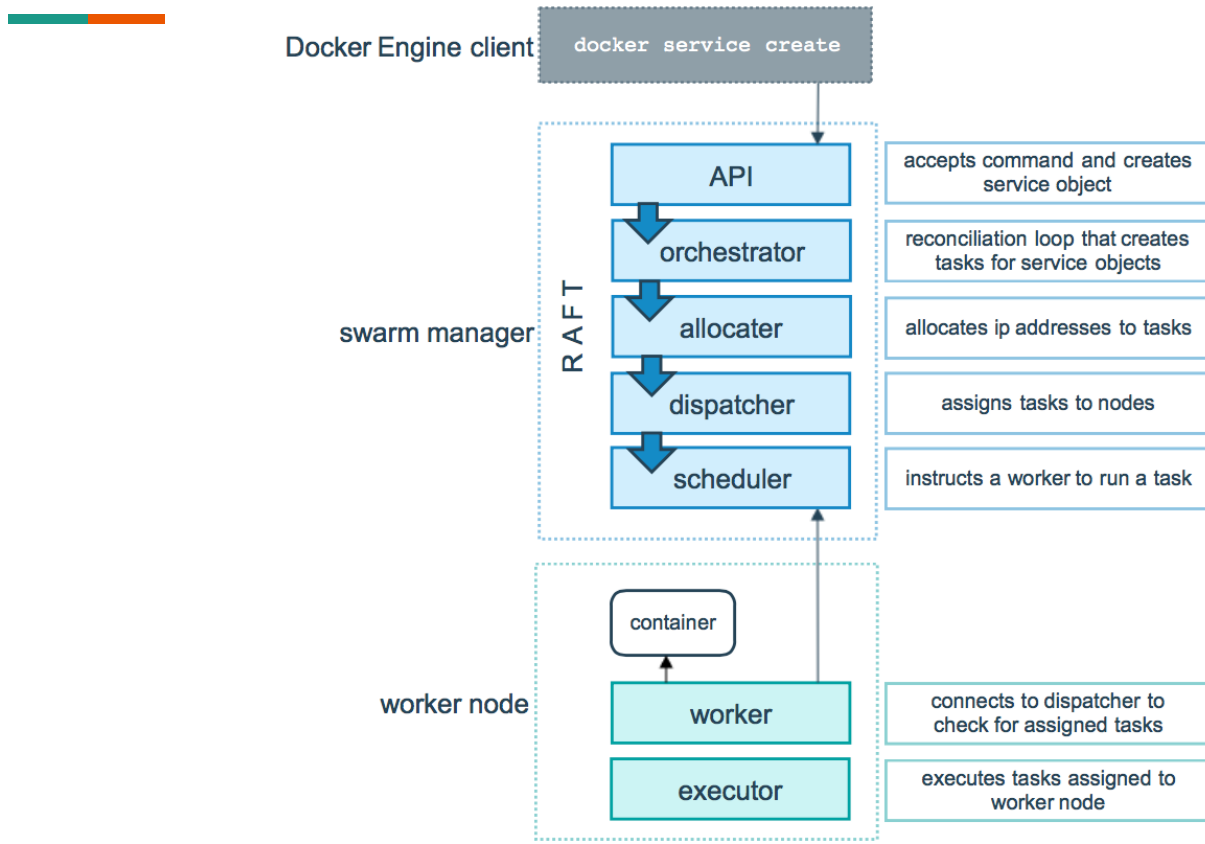
Replicas mode:

- **Replicated:** chỉ định số lượng tasks
- **Global services:** Đảm bảo mỗi Worker Node chạy một task





# Swarm Mode: Build-in orchestration





# Khởi tạo dịch vụ và mở rộng

Khởi tạo swarm:

```
$ docker swarm init --advertise-addr <MANAGER-IP>
```

*Swarm initialized: current node (dxn1zf6l61qsb1josja83ngz) is now a manager.*

*To add a worker to this swarm, run the following command:*

```
docker swarm join \  
--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxxv8rssmk743ojnwacrr2e7c \  
192.168.99.100:2377
```

*To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.*

**docker info** - kiểm tra trạng thái hiện tại của Swarm

**docker node ls** - kiểm tra thông tin của nodes



# Khởi tạo dịch vụ và mở rộng

Hiện thực một service trên Swarm:

```
docker service create --replicas 1 --name helloworld alpine ping 8.8.8.8
```

Docker service CLI:

`docker service ls` - danh sách các services

`docker service inspect --pretty <SERVICE-ID>` - hiển thị các thông tin chi tiết của service

`docker service ps <SERVICE-ID>` - kiểm tra các nodes đang chạy service

`docker service rm <SERVICE-ID>` - xóa service đang trên Swarm



# Khởi tạo dịch vụ và mở rộng

Mở rộng số lượng replica của service

```
docker service scale <SERVICE-ID>=<NUMBER-OF-TASKS>
```

Thực thi rolling updates trên service

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

```
docker service update --image redis:3.0.7 redis
```

Scheduler áp dụng rolling updates mặc định như sau:

- Dừng một task đầu tiên
- Cập nhật cho task vừa bị dừng
- Chạy lại container cho task vừa được cập nhật
- Nếu quá trình cập nhật trả về RUNNING, chờ một khoảng thời gian và tiếp tục chạy task tiếp theo
- Nếu trong quá trình cập nhật task trả về FAILED, dừng quá trình cập nhật

```
docker service update --rollback redis
```



# Khởi tạo 3 node-swarm cluster

Environment:

- Dùng browser: <https://labs.play-with-docker.com>
- Dùng docker-machine

Swarm node CLI: chỉ thực hiện được từ Manager node

`docker swarm join-token worker` - Tạo token để node tham gia là Worker

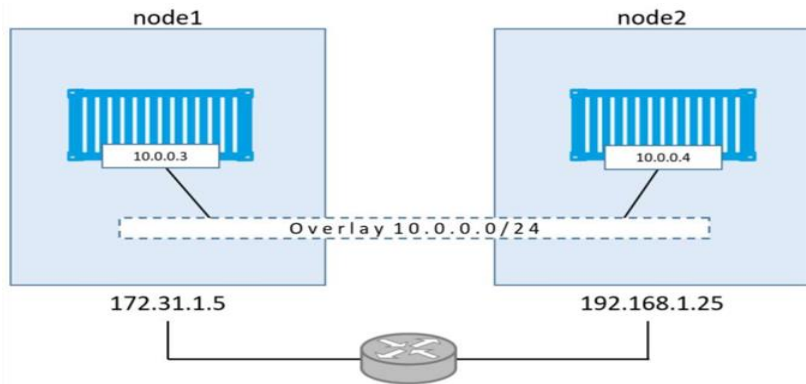
`docker swarm join-token manager` - Tạo token để node tham gia là Manager

`docker node promote <node-name>` - Nâng cấp node lên Manager

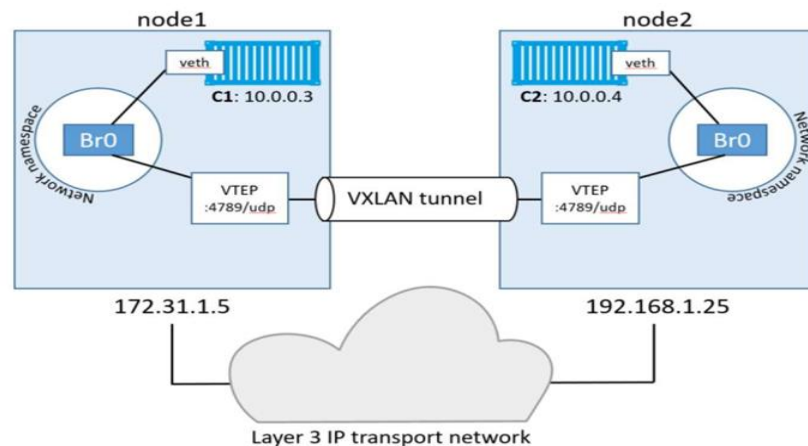


## Overlay Multi-Host Networking

- Chọn `--driver overlay` khi khởi tạo network
- Sử dụng cho giao tiếp giữa các container trong Swarm
- Có thể dùng IPsec (AES) để mã hóa
- Mỗi service có thể kết nối tới nhiều networks khác nhau (front-end, back-end)



Developer Edition



Network Guy Edition



# Routing Mesh

---

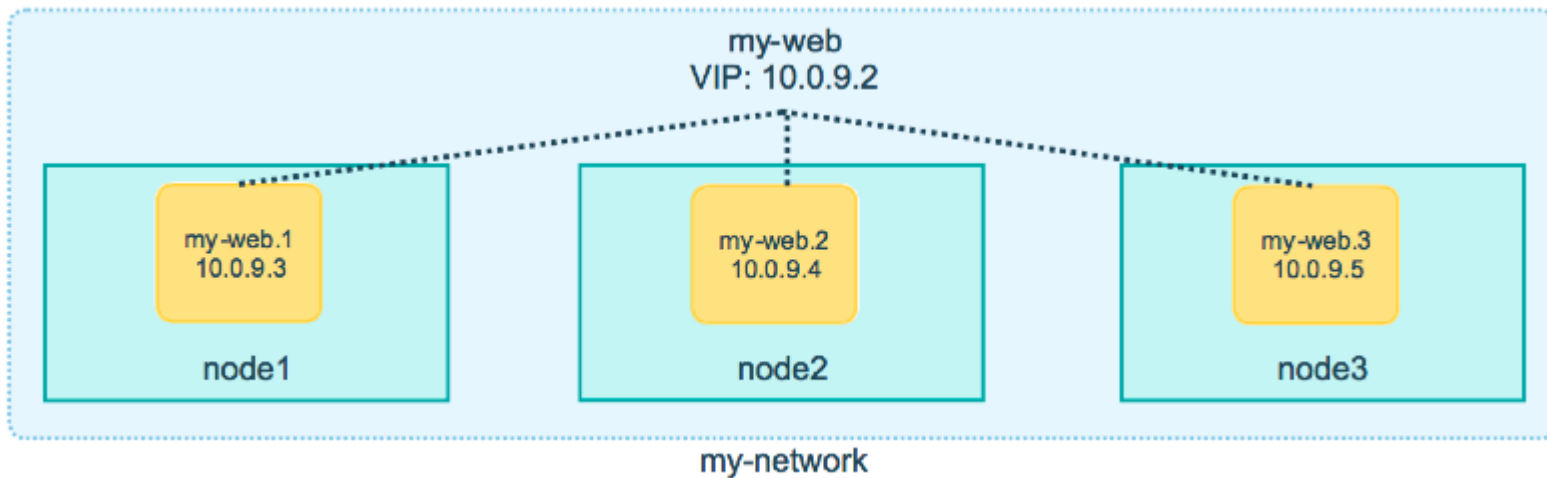
- Định tuyến các gói tin đi vào cho các service đến đúng task
- Mở rộng tất cả các node trong Swarm
- Load balances Swarm Services tới các Tasks, có hai cách thực hiện:
  - Giao tiếp giữa các Container trong Overlay Network (dùng VIP)
  - Kết nối từ ngoài vào tới các published port (tất cả các nodes)
- Là dạng stateless load balancing ở OSI Layer 3 (TCP), không phải Layer 4 (DNS)





# Routing Mesh

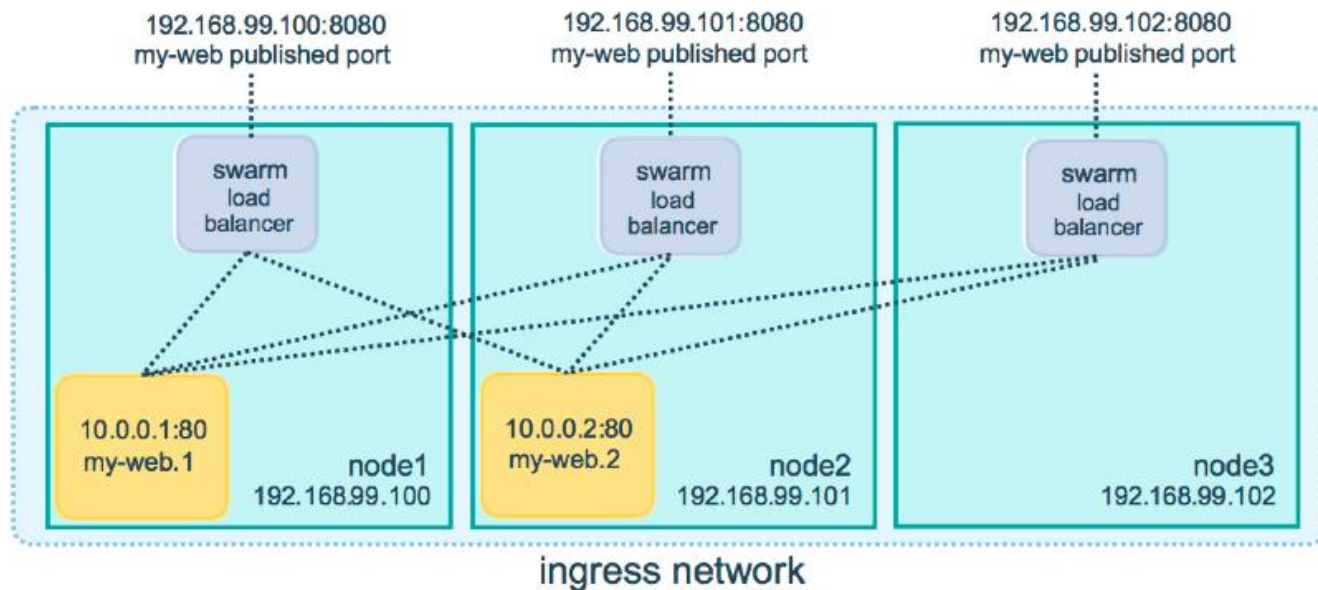
Giao tiếp giữa các Container trong Overlay Network (dùng VIP)





# Routing Mesh

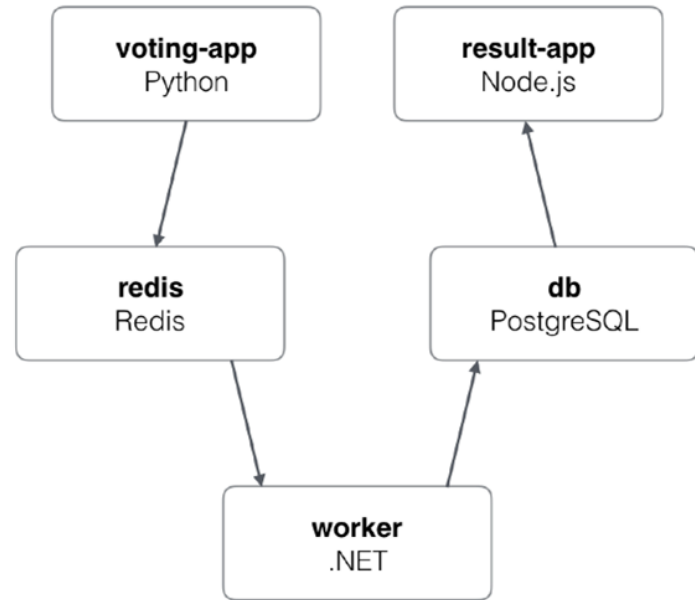
Kết nối từ ngoài vào tới các published port (tất cả các nodes)





# Assignment: Create Multi-Service App

- Sử dụng Docker Distributed Voting App
- Xem yêu cầu trong thư mục *swarm-app-1*
- Cần 1 volume, 2 networks and 5 services
- Cần dùng câu lệnh khởi tạo, chạy lên và kiểm tra các services







# Rancher



CONTAINER AUTO SCALING



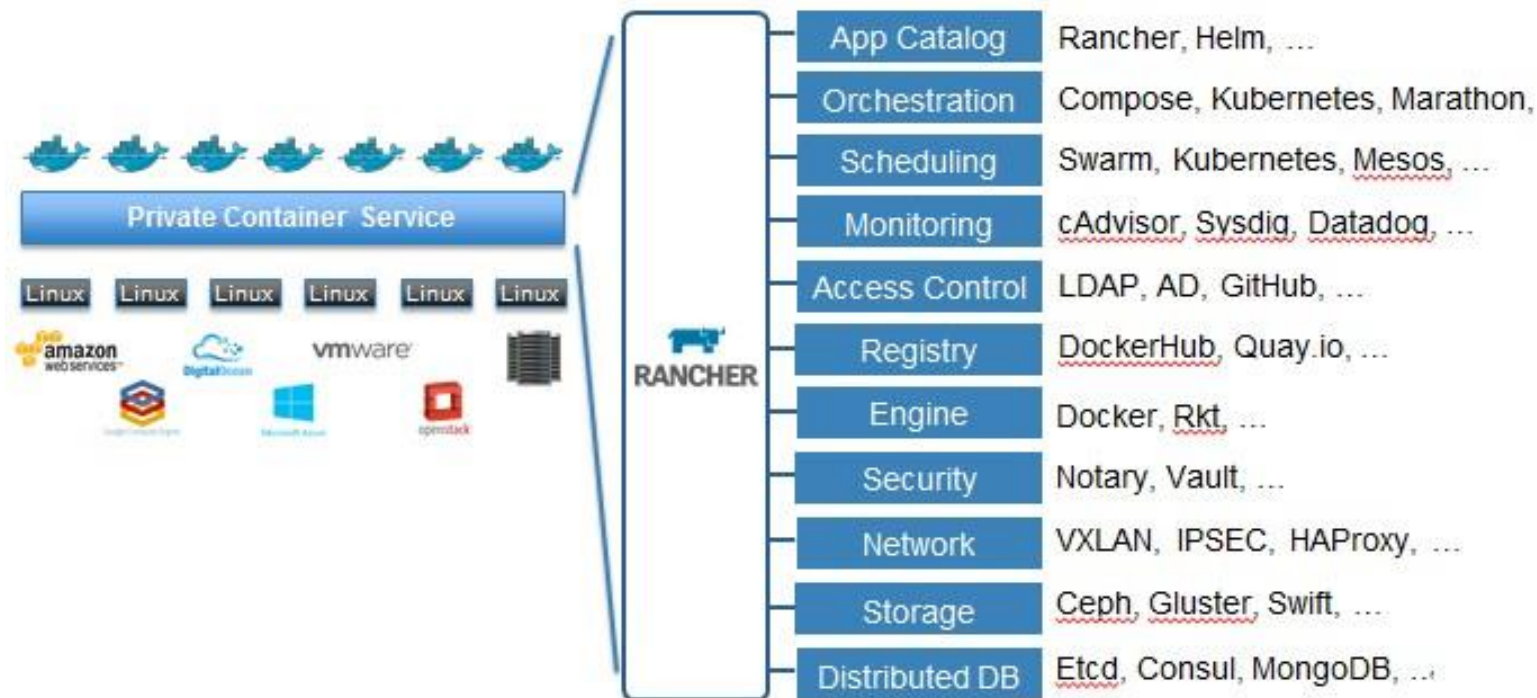
ON-PREM



INFRASTRUCTURE SCALING



# Rancher





# Docker on AWS

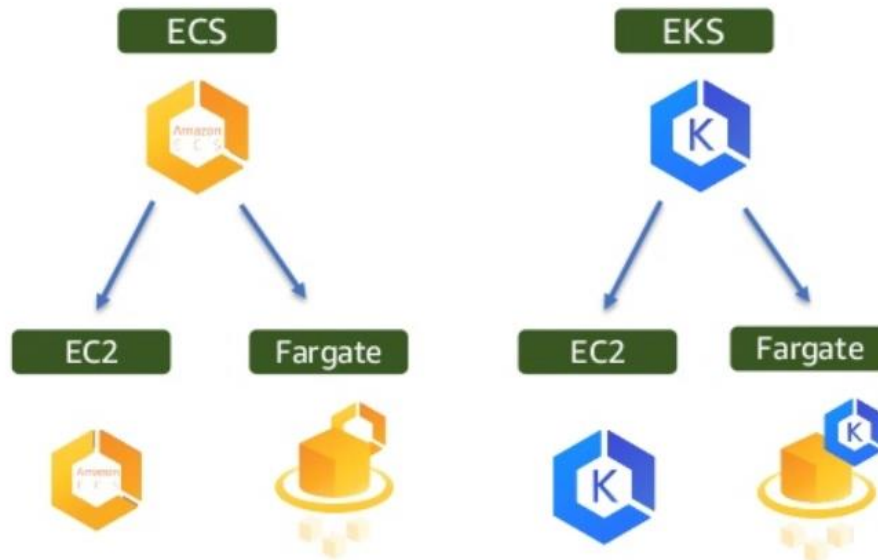
1

Choose your orchestration tool

2

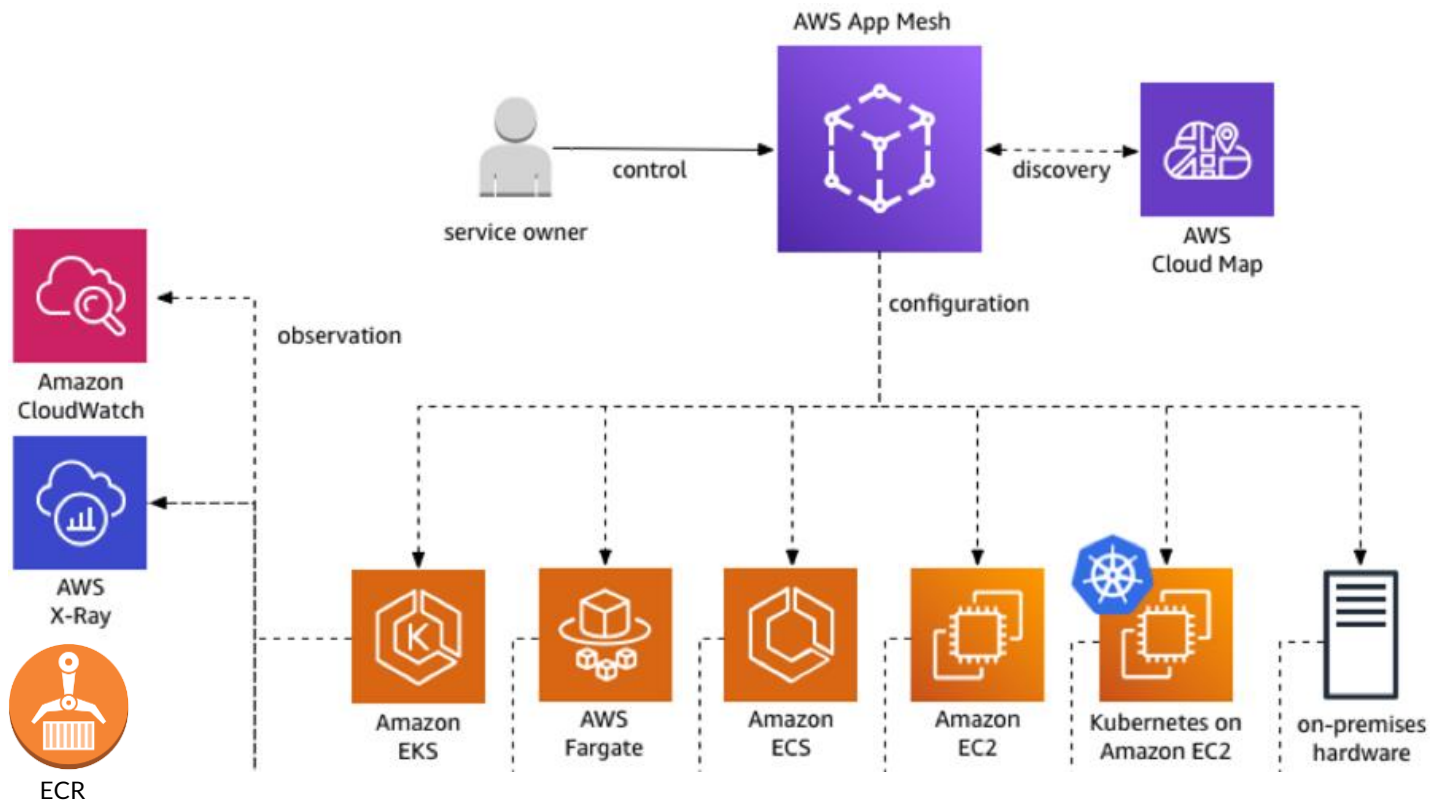
Choose your launch type

## AMAZON CONTAINER SERVICES





# Docker on AWS







# Raft - Understandable Distributed Consensus

## Docker Overlay Network