

Môn học

# DevOps

Giảng viên

**Tan Do**

0868880797



# Nội dung buổi 07

- Container orchestration
- Docker swarm orchestrator
- Tạo các máy chủ docker chạy bằng docker (docker:dind)
- Các câu lệnh cơ bản docker swarm
- Cài đặt và sử dụng docker swarm
- Khởi tạo service đầu tiên với docker swarm
- Docker stack là gì?
- Triển khai dịch vụ với docker stack
- Kiến trúc của docker swarm

# Container orchestration là gì

Với sự phát triển của các ứng dụng phức tạp và quy mô lớn, việc quản lý các container đơn lẻ trở nên không đủ, container orchestration ra đời nhằm giải quyết vấn đề này bằng cách

- Quản lý vòng đời của container
- Tự động hóa các quy trình triển khai, cấu hình và quản lý mạng

Điều phối container mang đến sự đột phá mạnh mẽ trong lĩnh vực công nghệ thông tin, mở ra cách cửa cho việc xây dựng và triển khai ứng dụng linh hoạt, hiệu quả và dễ dàng quản lý

# Các công cụ Container orchestration

Phổ biến:

- Docker Swarm
- Kubernetes
- Apache Mesos
- Nomad (Hashi Corp)

Cung cấp các tính năng quản lý mạnh mẽ, giúp tổ chức quản lý các ứng dụng container một cách dễ dàng và hiệu quả

# Lợi ích mô hình Container orchestration?

- Tạo và quản lý Container
- Thay đổi quy mô ứng dụng linh hoạt
- Quản lý vòng đời của Container
- Giám sát tình trạng của Container
- Tự động triển khai



# Docker swarm là gì?

- Là một công cụ giúp tạo ra một clustering Docker
- Gom nhiều Docker Engine lại với nhau và có thể “nhìn” nó như duy nhất một virtual Docker Engine
- Bất cứ công cụ nào có thể giao tiếp với Docker Engine thì cũng sẽ giao tiếp với Docker Swarm bình thường theo đúng chuẩn Docker API và tất nhiên là trong suốt
- Một cụm Swarm có thể được cấu hình và deploy thông qua Docker-Machine

# Tạo các máy chủ docker chạy bằng docker

- Docker in Docker (docker:dind)
- docker pull docker:dind
- docker run -it --privileged --name docker -d docker:dind
  - --privileged: Cho phép container này có quyền truy cập đầy đủ vào các thiết bị của host, điều này cần thiết để Docker bên trong container có thể hoạt động đúng
  - --name docker: Đặt tên container là mydocker (tên khác tùy ý)
  - -d: Chạy container ở chế độ nền (detached mode)
- Truy cập vào container docker:dind
  - docker exec -it docker sh
  - cat /etc/os-release

# Tạo các máy chủ docker chạy bằng docker

- Cài đặt bên trong docker:dind
  - apk update
  - apk add docker
  - rc-update add docker boot
  - service docker start
    - apk update: Cập nhật các gói trước khi cài đặt
    - apk add docker: Cài đặt Docker
    - rc-update add docker boot: Thêm Docker vào danh sách các dịch vụ tự động khởi động khi boot
    - service docker start: Khởi động Docker service
- Tạo ít nhất 2 máy chủ docker:dind để thực hành docker swarm
  - docker1
  - docker2



# Các câu lệnh cơ bản docker swarm

- docker swarm
- docker node
- docker service
- docker stack
- docker secret

# Cài đặt và sử dụng docker swarm

- Kiểm tra trạng thái
  - docker info
  - docker info | grep Swarm
- Kích hoạt docker swarm
  - docker swarm init
    - To add a worker to this swarm, run the following command:
      - docker swarm join --token SWMTKN-1-5usz3tpha4ibhm4cy5jmuz3rg3skqfd3ah7na3cn85f1jqs51g-aq8nnqfsseo28gzequtca3xq5192.168.65.3:2377
    - To add a manager to this swarm, run:
      - docker swarm join-token manager
    - Lấy lại token
      - docker swarm join-token worker
  - docker swarm leave

# Khởi tạo service đầu tiên với docker swarm

- docker info
- docker info | grep Swarm   # **g**lobal **r**egular **e**xpression **p**rint
- docker node ls
- docker node --help
  - docker node rm node-name
- docker service --help
- docker service create alpine ping 8.8.8.8
- docker service ls
- docker service ps [service name]

# Khởi tạo service đầu tiên với docker swarm

- Kiểm tra dịch vụ đã tạo
  - docker service ls
  - docker service ps [service name]
- Tạo dịch vụ webserver
  - docker service create --name webserver --replicas 3 nginx
  - docker service ls
  - docker service ps webserver
- Mở rộng dịch vụ
  - docker service scale webserver=10
- Cập nhật cấu hình cluster
  - docker service update --image=nginx:latest webserver
  - docker service update webserver --replicas 5
- Rollback
  - docker service rollback webserver
- Xóa một cluster
  - docker service rm webserver
- Theo dõi
  - docker stats

# Khởi tạo service đầu tiên với docker swarm

- Nâng cấp 1 node lên manager
  - Vào node manager
  - docker node ls
  - docker node promote node-id
- Hạ cấp 1 node khỏi manager
  - Vào node manager
  - docker node ls
  - docker node demote node-id
- Thực hành thêm với --help



# Khởi tạo service đầu tiên với docker swarm

- Kiểm tra network trên các host trong Cluster
  - docker network ls
  - docker inspect [id\_container]
- Để các container có thể giao tiếp được với nhau, chạy lệnh thêm cờ -p hoặc cờ --network ingress khi khởi tạo. Mạng ingress là mạng mặc định được tạo ra khi khởi tạo một cluster và có phạm vi là swarm
  - docker service create --replicas 5 -t -p 22:22 --network ingress --name server busybox

# Docker stack là gì?

## - Docker Swarm:

- Đơn thuần điều phối các container chạy trên nhiều các docker host.
- Nhưng giữa các container đây không có sự phụ thuộc lẫn nhau
- Để liên kết các container thành một ứng dụng cần phải thao tác thủ công cho phép kết nối giữa chúng

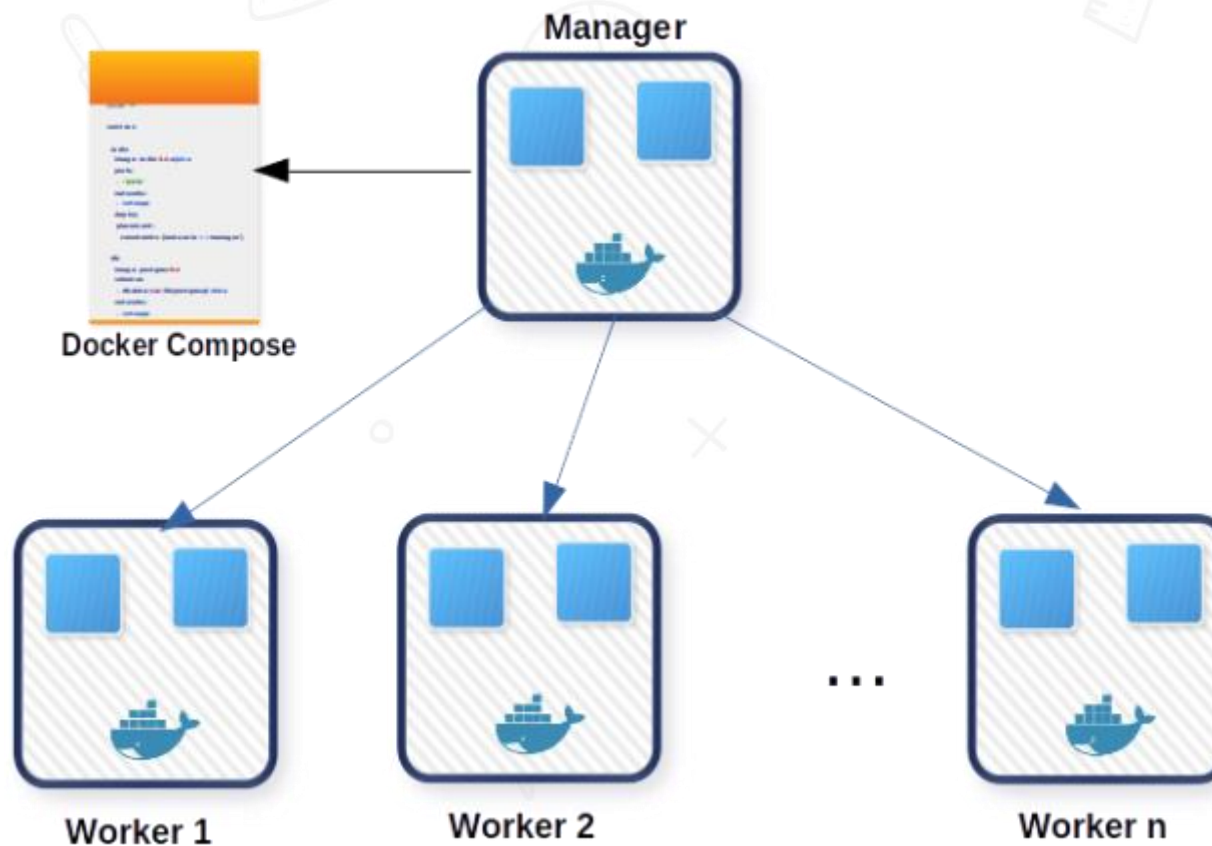
## - Docker Compose:

- Là công cụ mà cho phép liên kết các container với nhau để chạy ứng dụng
- Nhưng các container đây lại chạy trên cùng một docker host
- Vì vậy, sẽ có mặt hạn chế khi muốn high availability và horizontal scaling

## - Docker Stack:

- Cho phép chúng ta kết nối 2 công nghệ trên với nhau
- Sử dụng tệp tin docker-compose (mặc định docker-compose.yml) để định nghĩa, liên kết các container với nhau và chạy trong một cluster docker host (docker swarm)

# Docker stack là gì?



# Triển khai dịch vụ với Docker stack

Các bước triển khai:

- Tạo file docker-compose.yml
- Triển khai dịch vụ
- Kiểm tra

# Triển khai dịch vụ với Docker stack

- File docker-compose.yml

```
name: myapp
services:
  nginx:
    image: nginx
    ports:
      - 80:80
    networks:
      - nw1
    volumes:
      - ./var/www/html
    deploy:
      replicas: 3
      resources:
        limits:
          cpus: '0.75' # 75%
          memory: 100MB
    networks:
      nw1:
```



# Triển khai dịch vụ với Docker stack

- Deploy dịch vụ
  - docker stack deploy --compose-file docker-compose.yml webserver
- Kiểm tra trạng thái
  - docker stack ls
  - docker stack services webserver\_nginx
- Xóa stack
  - docker stack rm webserver\_nginx

# Triển khai dịch vụ với Docker stack

Các bước triển khai:

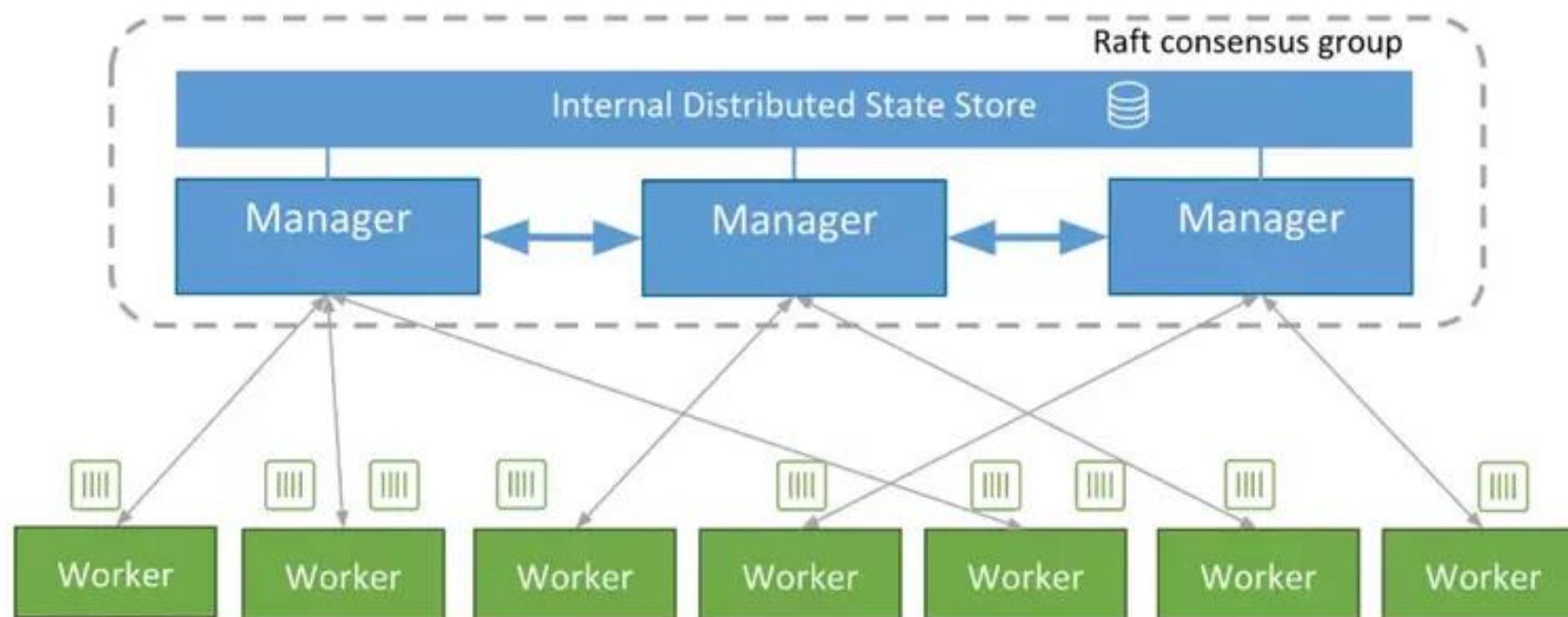
- Tạo file docker-compose.yml
- Triển khai dịch vụ
- Kiểm tra

# Kiến trúc Docker Swarm

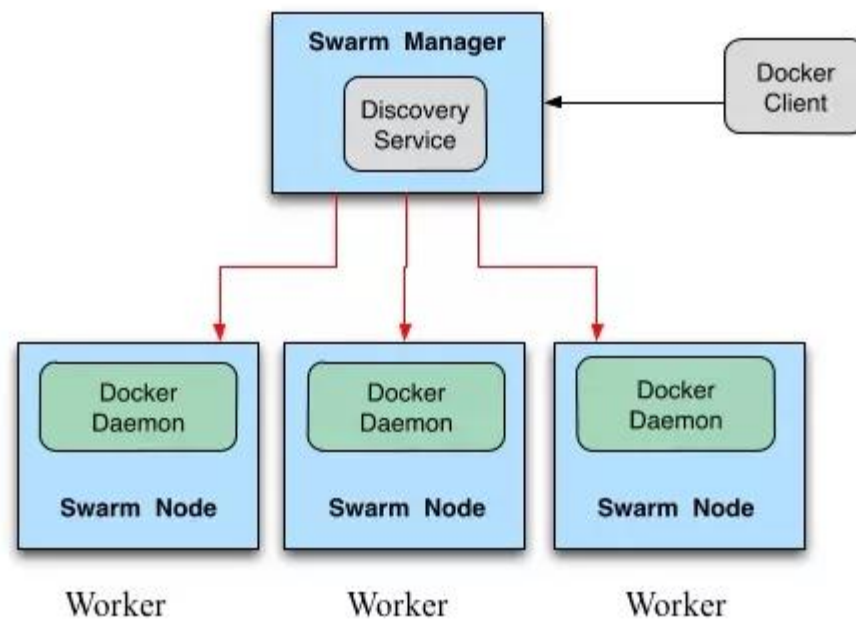
Các thành phần chính:

- Swarm:
  - Một tập hợp các node có ít nhất một nút chính và một số node worker có thể là máy ảo hoặc vật lý
- Service:
  - Các tasks được xác định bởi người quản trị swarm mà người quản lý hoặc các node agent(đại lý) phải thực hiện
  - Nó xác định hình ảnh container nào nên sử dụng và lệnh nào sẽ chạy trong mỗi container
  - Nó xác định container images nào mà swarm nên sử dụng và lệnh nào swarm sẽ chạy trong mỗi container
- Manager Node:
  - Node manager cung cấp một số chức năng sau khi deploy một ứng dụng, chẳng hạn như:
    - Nó phân phối công việc (dưới dạng nhiệm vụ) cho các node worker
    - Quản lý trạng thái của swarm mà nó thuộc về
- Worker Node:
  - Các node worker chạy các tác vụ được phân phối bởi node manager trong swarm
  - Mỗi node worker chạy một agent (tác nhân) báo cáo lại cho node master về trạng thái của các tác vụ được gán cho nó, vì vậy node manager có thể theo dõi các dịch vụ và tác vụ đang chạy trong swarm
- Task:
  - Tasks là các Docker containers thực thi các lệnh đã định nghĩa trong service
  - Các node manager gán các task cho các node worker, và sau khi việc gán này, task không thể chuyển sang một worker khác. Nếu task thất bại trong bộ bản sao, người quản lý sẽ chỉ định một phiên bản mới của tác vụ đó cho một node có sẵn khác trong swarm

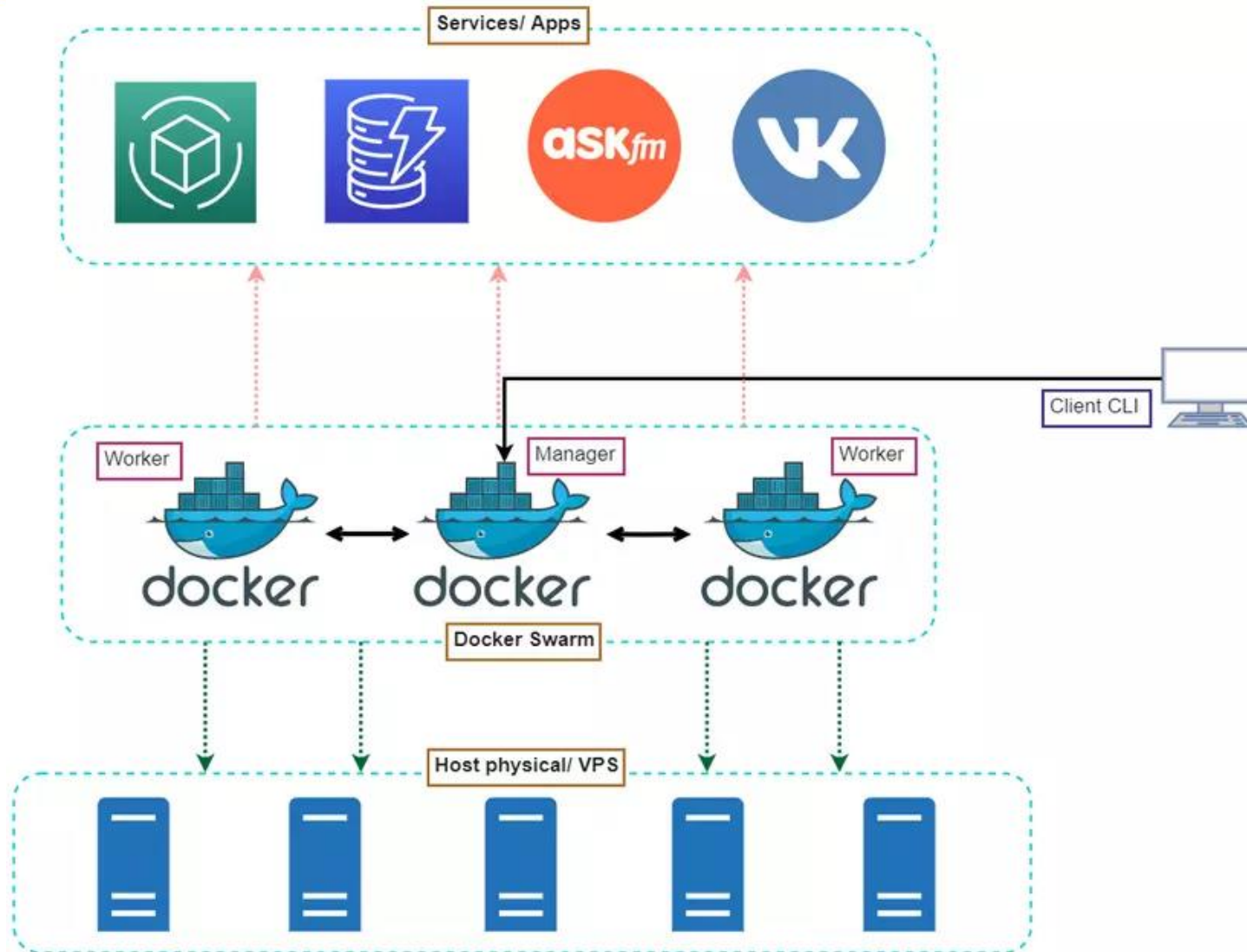
## Swarm Architecture



# Kiến trúc Docker Swarm

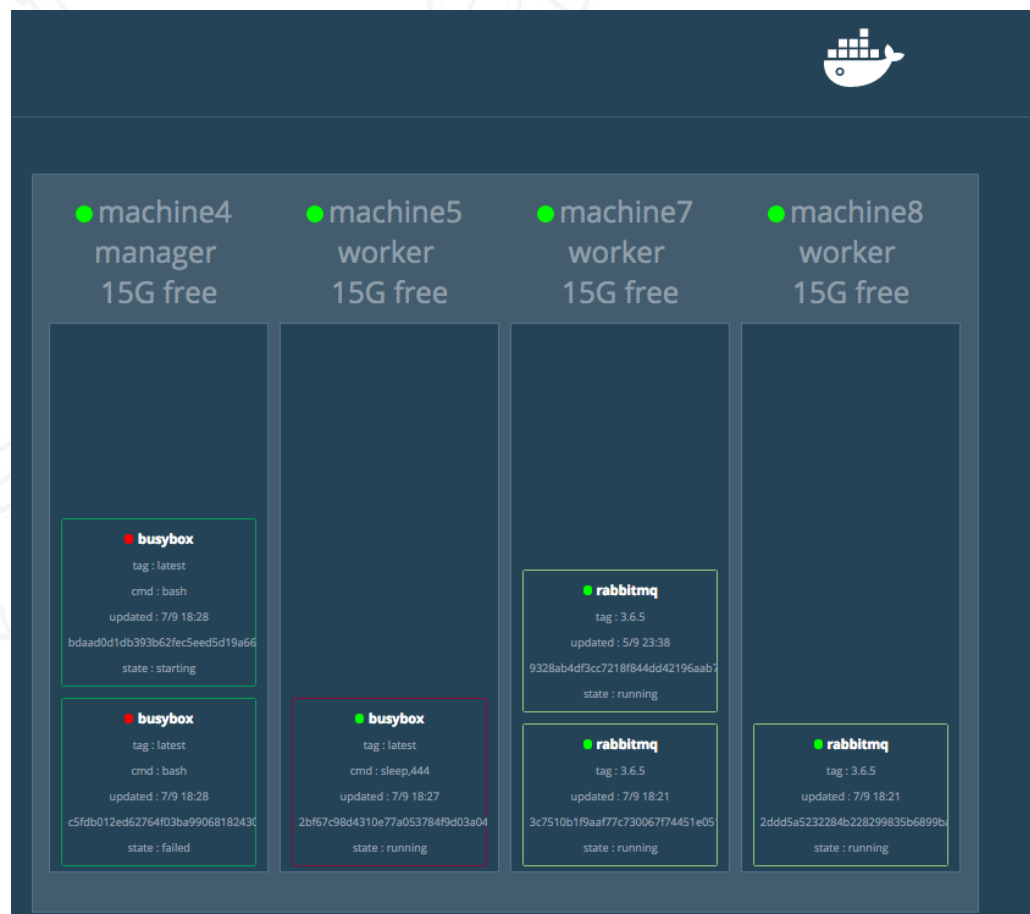






# Docker Swarm Visualizer

- <https://github.com/dockersamples/docker-swarm-visualizer>



# THANK YOU