**Phần 19 Docker Swarm**

Docker Swarm là một công cụ nhóm và sắp xếp gốc giúp quản lý các công cụ docker. Trong đó, nhóm các công cụ docker được chuyển thành một máy chủ docker ảo duy nhất. Nó giúp người dùng triển khai, quản lý và mở rộng quy mô các ứng dụng một cách liền mạch trên nhiều nút docker. Nó đi kèm với việc cung cấp các tính năng như khám phá dịch vụ, cân bằng tải, mở rộng quy mô và cập nhật liên tục. Nó cung cấp một cách dễ dàng và tích hợp để quản lý các ứng dụng được chứa trong một cụm.

**Nhiệm vụ và dịch vụ trong Docker Swarm là gì?**

Trong Docker Swarm, **Service** là một khái niệm trừu tượng cấp cao hơn được sử dụng để xác định cách các container nên được triển khai, quản lý và mở rộng trên một nhóm các nút Docker. Nó bao gồm các thông số kỹ thuật như số lượng bản sao, mạng và cân bằng tải.

Một **Task** là một trường hợp riêng lẻ của một container được tạo ra và quản lý bởi một dịch vụ. Mỗi task đại diện cho một đơn vị công việc duy nhất và được lên lịch chạy trên một trong các nút trong swarm. Task là các container đang chạy thực tế đáp ứng các yêu cầu do dịch vụ chỉ định.

**Docker Swarm hoạt động như thế nào?**

Khi bạn muốn triển khai một container trong swarm trước tiên, bạn phải khởi chạy các dịch vụ. Dịch vụ bao gồm nhiều container của cùng một hình ảnh. Các dịch vụ này được triển khai bên trong một nút nên để triển khai một swarm, ít nhất một nút phải được triển khai. Như bạn thấy trong sơ đồ bên dưới, nút quản lý chịu trách nhiệm phân bổ tác vụ, phân phối tác vụ và lên lịch cho các tác vụ. [API](https://www.geeksforgeeks.org/what-is-an-api/) trong trình quản lý là phương tiện giữa nút quản lý và nút công nhân để giao tiếp với nhau bằng cách sử dụng [giao thức HTTP](https://www.geeksforgeeks.org/types-of-internet-protocols/) . Dịch vụ của một cụm có thể được cụm kia sử dụng. Tất cả các thực thi tác vụ đều được thực hiện bởi nút công nhân.

**Khởi tạo Docker Swarm**

Để khởi tạo cụm Docker Swarm, chúng ta sử dụng lệnh có tên là **“docker swarm init”.** Để chuyển đổi docker engine thành swarm manager, chúng ta sẽ sử dụng lệnh này sau khi chuyển đổi sang chế độ swarm, sau đó bạn sẽ có thể quản lý và các nút worker, sau đó swarm sẽ phân phối công việc cho chúng.

**Sau khi khởi tạo chế độ Swarm**

* Sau khi khởi tạo docker swarm init, nó sẽ đưa swarm vào hoạt động.
* Sau khi đưa bầy đàn vào hoạt động, hãy biết rằng nó sẽ biến nút hiện tại thành nút quản lý. Và nó cũng sẽ tạo ra một mã thông báo.
* Mã thông báo được tạo ra sẽ được sử dụng để tiếp tục kết nối các nút công nhân và nút chính.

**Tính năng Docker Swarm**

**Cluster management integrated with Docker Engine:** Quản lý cluster với Docker Engine bằng việc sử dụng Docker CLI để tạo swarm.

**Decentralized design:** Docker Swarm được thiết kế dạng phân cấp. Thay vì xử lý sự khác biệt giữa các roles của node tại thời điểm triển khai, Docker Engine xử lý bất kỳ chuyên môn hoá nào khi runtime. Bạn có thể triển khai cả hai loại node: managers và worker bằng Docker Engine.

**Declarative service model:** Docker Engine sử dụng phương thức khai báo để cho phép bạn define trạng thái mong muốn của các dịch vụ khác nhau trong stack ứng dụng của bạn. VD: Bạn có thể mô tả ứng dụng bao gồm: web front-end với service message queueing và database back-end.

**Scaling:** Đối với mỗi service bạn có thể khai báo số lượng task mà bạn muốn run. Khi bạn scale up hoặc down thì swarm manager sẽ tự động thêm hoặc xoá task để duy trì trạng thái mong muốn.

**Desired state reconciliation:** Hãy hình dung bạn thiết lập một service run 10 replicas của một container và một worker machine (host/vps) đang giữ 2 trong số 10 replicas đó gặp sự cố bị crash, lúc này swarm manager sẽ tiến hành tạo thêm 2 replicas mới để thay để cho 2 replicas đã bị crash đó và tiến hành chuyển 2 replicas mới này cho các worker đang run.

**Multi-host networking:** Bạn có thể chỉ định một overlay network cho các services của mình. Swarm manager sẽ tự động gán địa chỉ IP cho các container trên overlay network khi nó khởi tạo và cập nhật application.

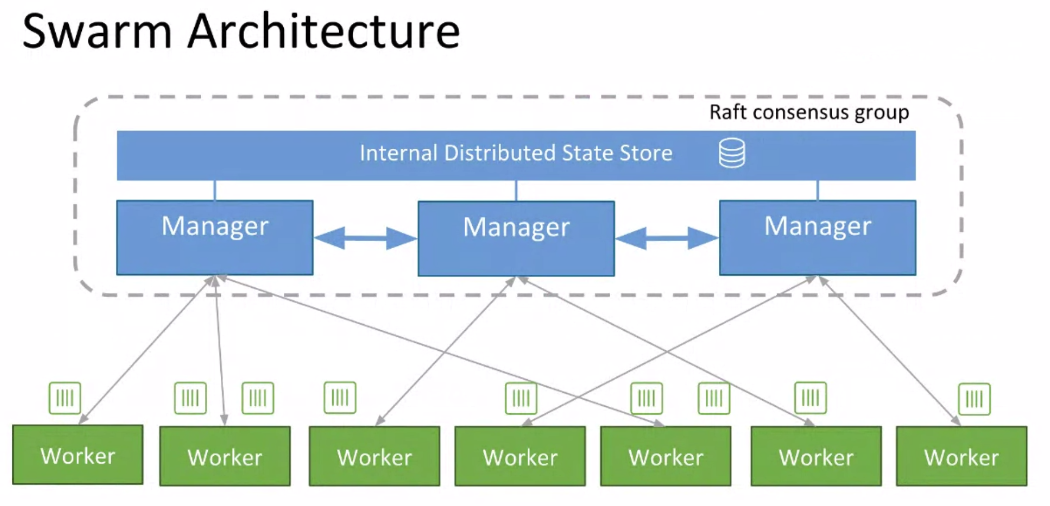
**Service discovery:** Swarm manager node gán mỗi service trong swarm một DNS duy nhất và bạn có thể truy vấn được thông qua DNS này.

**Load balancing:** Có thể expose các port cho các services tới load balance để giao tiếp với bên ngoài.

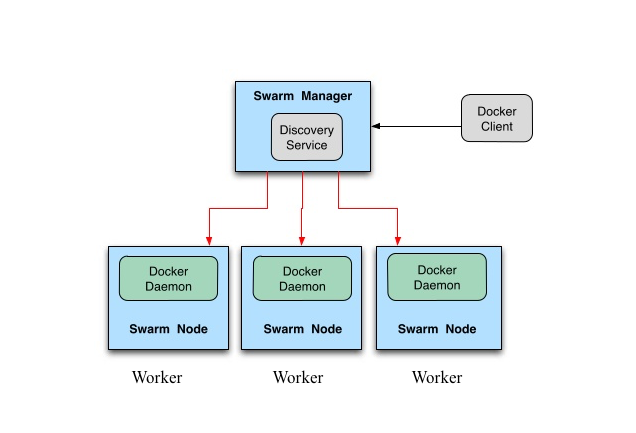
**Secure by default:** Các service giao tiếp với nhau thông qua giao thức bảo mật TLS. Bạn có thể tuỳ chỉnh sử dụng chứng chỉ ký tự root hoặc chứng chỉ từ một custom root CA.

**Rolling updates:** Swarm giúp bạn update image của service một cách hoàn toàn tự động. Swarm manager giúp bạn kiểm soát độ trễ giữa service deploy tới các node khác nhau và bạn có thể rolling back bất cứ lúc nào.

**Kiến trúc Swarm**

****

Bao gồm các Manager và các Worker. Người dùng có thể khai báo trạng thái mong muốn của nhiều service để chạy trong Swarm sử dụng YAML files.



* Swarm: là một cluster của một hoặc nhiều Docker Engine đang run (cụ thể ở đây là các node) trong chế độ Swarm, thay vì phải chạy các container bằng câu lệnh thì ta sẽ thiết lập các services để phân bổ các bản replicas tới các node.
* Node: Một node là một máy vật lý hay máy ảo đang run phiên bản Docker Engine trong chế độ Swarm. Node sẽ gồm hai loại: Manager Node và Worker Node.
* Manager Node: Là node nhận các define service từ user, nó quản lý và điều phối các task đến các node Worker. Theo mặc định node Manager cũng được coi là node Worker.
* Worker Node: là node nhận và thực thi các task từ node Manager.
* Service: Một service xác định image của container và số lượng các replicas (bản sao) mong muốn khởi chạy trong swarm.
* Task: là một tác vụ mà node worker phải thực hiện. Tác vụ này sẽ do node Manager phân bổ xuống. Một task mang một Docker Container và các lệnh để chạy bên container.

**Docker Swarm được sử dụng để làm gì?**

Docker Swarm là một công cụ điều phối container được sử dụng để Docker container và mở rộng chúng. Thay vì một máy chủ duy nhất với sự trợ giúp của Docker Swarm, chúng ta có thể quản lý nhiều nút được gọi là cụm, nơi chúng ta có thể triển khai và duy trì container của mình trong nhiều máy chủ.

1. Docker Swarm đảm bảo rằng ứng dụng của chúng tôi luôn khả dụng ngay cả khi một trong các nút bị lỗi bằng cách tạo vùng chứa trong một nút khác khả dụng
2. Dựa trên lưu lượng truy cập đến, chúng ta có thể tăng hoặc giảm quy mô container bằng cách thêm vào nhiều nút.
3. Nếu có nhiều container, lượng tải đến sẽ được Docker Swarm tự động cân bằng.
4. Docker Swarm có một số tính năng bảo mật, bao gồm mã hóa lưu lượng giữa các nút và xác thực TLS lẫn nhau.
5. Docker Swarm sẽ tự động xử lý các container và node bị lỗi. Nhờ đó, chúng ta có thể duy trì tính khả dụng cao.

Simply Docker Swarm chủ yếu được sử dụng để triển khai, mở rộng quy mô và quản lý các container và nút có sẵn trong cụm.

**Các chế độ khác nhau của Docker Swarm**

Docker Swarm chủ yếu bao gồm hai chế độ:

1. **Chế độ toàn cục:** Ở chế độ này, Docker Swarm sẽ duy trì các container trong tất cả các nút slave và nút master. Nó sẽ duy trì các bản sao của container trong tất cả các nút có sẵn trong cụm.
2. **Chế độ sao chép:** Ở chế độ này, Docker Swarm sẽ triển khai các container dựa trên số lượng bản sao cần thiết cho bạn. Nếu bạn yêu cầu 3 bản sao, nó sẽ triển khai các container dựa trên tính khả dụng của nút.