I. Khái niệm

-Docker là một nền tảng cho lập trình viên và quản trị hệ thống để phát triển, triển khai và chạy ứng dụng bằng container;

-Docker là một open platform cung cấp cho người sử dụng những công cụ và service để người sử dụng có thể đóng gói và chạy chương trình của mình trên các môi trường khác nhau một cách nhanh nhất;

-Docker là một nền tảng mở dành cho các lập trình viên, quản trị hệ thống dùng để xây dựng, vận chuyển và chạy các ứng dụng phân tán;

II. Ưu điểm

+Tiện lợi: Bình thường khi cần chạy ứng dụng chúng ta cần cài đầy đủ môi trường trên máy tính, chưa kể có sự xung đột, sự cố xảy ra với các ứng dụng. Với Docker, bạn có thể đóng gói tất cả các thành phần của ứng dụng vào một container và chạy nó trên bất kỳ máy tính nào mà không cần phải cài đặt lại môi trường.

+Dễ dàng sử dụng: Docker rất dễ cho mọi người sử dụng từ developers, systems admins, architects… nó tận dụng lợi thế của container để build, test nhanh chóng. Có thể đóng gói ứng dụng trên laptop của họ và chạy trên public cloud, private cloud… Câu thần chú là “Build once, run anywhere”.

+Tốc độ: Docker Containers tương đối nhẹ và có tốc độ rất nhanh. Bạn hoàn toàn có thể tạo và khởi chạy chỉ trong vài giây.

+Linh hoạt: Triển khai ở bất kỳ nơi đâu do sự phụ thuộc của ứng dụng vào tầng OS cũng như cơ sở hạ tầng được loại bỏ.

+Môi trường chạy và khả năng mở rộng: Bạn có thể chia nhỏ những chức năng của ứng dụng thành các container riêng lẻ. Ví dụng Database chạy trên một container và Redis cache có thể chạy trên một container khác trong khi ứng dụng Node.js lại chạy trên một cái khác nữa. Với Docker, rất dễ để liên kết các container với nhau để tạo thành một ứng dụng, làm cho nó dễ dàng scale, update các thành phần độc lập với nhau.

III. Các thành phần cơ bản

* + Docker Engine: dùng để tạo ra Docker image và chạy Docker container. Là thành phần chính của Docker, như một công cụ để đóng gói ứng dụng.
  + Docker Hub: dịch vụ lưu trữ giúp chứa các Docker image. Trên DockerHub có hàng ngàn public images được tạo bởi cộng đồng cho phép bạn dễ dàng tìm thấy những image mà bạn cần. Và chỉ cần pull về và sử dụng với một số config mà bạn mong muốn.
  + Docker Image: một dạng tập hợp các tệp của ứng dụng, được tạo ra bởi Docker engine. Nội dung của các Docker image sẽ không bị thay đổi khi di chuyển. Docker image được dùng để chạy các Docker container. Bạn có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẽ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile.
  + Docker Container: một dạng runtime của các Docker image, dùng để làm môi trường chạy ứng dụng. Bạn có thể create, start, stop, move or delete container dựa trên Docker API hoặc Docker CLI.
  + Dockerfile: là một tập tin bao gồm các chỉ dẫn để build một image.
  + Volumes: là một cơ chế được Docker sử dụng để cung cấp khả năng lưu trữ liên tục (persistent data storage) cho các container (tránh mất dữ liệu khi container bị xóa). Docker Volume là một tính năng quan trọng giúp quản lý dữ liệu của container một cách dễ dàng và đảm bảo tính độc lập giữa container và máy chủ2
  + Docker Registry: là nơi lưu trữ riêng của Docker Images. Images được push vào registry và client sẽ pull images từ registry. Có thể sử dụng registry của riêng bạn hoặc registry của nhà cung cấp như : Alibaba Cloud, AWS, Google Cloud, Microsoft Azure...Docker Networking: cho phép kết nối các container lại với nhau. Kết nối này có thể trên 1 host hoặc nhiều host.
  + Docker Repository: là tập hợp các Docker Images cùng tên nhưng khác tags. VD: golang:1.11-alpine.
  + Docker Machine: tạo ra các Docker engine trên máy chủ.
  + Docker Client: là một công cụ giúp người dùng giao tiếp với Docker host thông qua command.
  + Docker Daemon: lắng nghe các yêu cầu từ Docker Client để quản lý các đối tượng như Container, Image, Network và Volumes thông qua REST API. Các Docker Daemon cũng giao tiếp với nhau để quản lý các Docker Service.
  + Docker Compose: là công cụ cho phép run app với nhiều Docker containers 1 cách dễ dàng hơn. Docker Compose cho phép bạn config các command trong file docker-compose.yml để sử dụng lại. Có sẵn khi cài Docker.
  + Docker Swarm: Docker Swarm là một công cụ trong hệ sinh thái Docker được sử dụng để quản lý và điều phối các container. Nó cho phép bạn triển khai và quản lý các ứng dụng dựa trên Docker trên một cụm (cluster) các máy chủ. Dưới đây là những điểm cơ bản về Docker Swarm:
    - Cụm (Cluster) Docker: Docker Swarm cho phép nhóm các máy chủ vật lý hoặc ảo có cài đặt Docker daemon thành một cụm (cluster). Bạn có thể xem nó như là một docker host ảo (virtual Docker host) duy nhất. Cụm này giúp tổ chức và quản lý các container một cách hiệu quả.
    - Tính Năng Của Docker Swarm:
      * -Quản lý cluster với Docker Engine: Sử dụng Docker CLI để tạo swarm và quản lý cụm với Docker Engine.
      * -Thiết kế phân cấp (Decentralized design): Docker Swarm được thiết kế để xử lý bất kỳ chuyên môn hoá nào khi runtime, cho phép triển khai cả hai loại node: managers và workers bằng Docker Engine.
      * -Mô hình dịch vụ khai báo (Declarative service model): Định nghĩa trạng thái mong muốn của các dịch vụ trong ứng dụng của bạn bằng phương thức khai báo.
      * -Tăng cường tỷ lệ (Scaling): Khai báo số lượng task cho mỗi dịch vụ và swarm manager tự động thêm hoặc xoá task để duy trì trạng thái mong muốn.
      * -Đồng bộ trạng thái mong muốn (Desired state reconciliation): Tự động thay thế các replicas bị crash bằng replicas mới.
      * -Mạng đa máy chủ (Multi-host networking): Chỉ định overlay network cho các services và swarm manager tự động gán địa chỉ IP cho các container.
      * -Phát hiện dịch vụ (Service discovery): Gán DNS duy nhất cho mỗi service trong swarm.
      * -Cân bằng tải (Load balancing): Expose các port cho các services để giao tiếp với bên ngoài.
      * -Bảo mật mặc định (Secure by default): Giao tiếp qua giao thức bảo mật TLS.
      * -Cập nhật dự phòng (Rolling updates): Tự động cập nhật image của service.
  + Docker Services: là các containers trong production. 1 service chỉ run 1 image nhưng nó mã hoá cách thức để run image — sử dụng port nào, bao nhiêu bản sao container run để service có hiệu năng cần thiết và ngay lập tức.
  + Docker Object: khi sử dụng docker, bạn có thể khở tạo hoặc sử các images, container, network, volumes, plugins hoặc các object khác. Những thành phần này được gọi chung là docker objects.
* Sự khác biệt giữa Docker Images và Docker Containers
  + Docker Images: Là một template chỉ cho phép đọc, ví dụ một image có thể chứa hệ điều hành Ubuntu và web app. Images được dùng để tạo Docker container. Docker cho phép chúng ta build và cập nhật các image có sẵn một cách cơ bản nhất, hoặc bạn có thể download Docker images của người khác.
  + Docker Containers: Docker container có nét giống với các directory. Một Docker container giữ mọi thứ chúng ta cần để chạy một app. Mỗi container được tạo từ Docker image. Docker container có thể có các trạng thái run, started, stopped, moved và deleted.

IV. Cách build docker container

1.Tạo Dockerfile:

\*Tạo một file có tên Dockerfile.

\*Trong Dockerfile, bạn định nghĩa các bước để xây dựng image. Ví dụ:

2.Build Docker image:

Mở terminal và di chuyển đến thư mục chứa Dockerfile.

Sử dụng lệnh sau để build image (thay đổi tên image theo ý muốn):

3.Kiểm tra image:

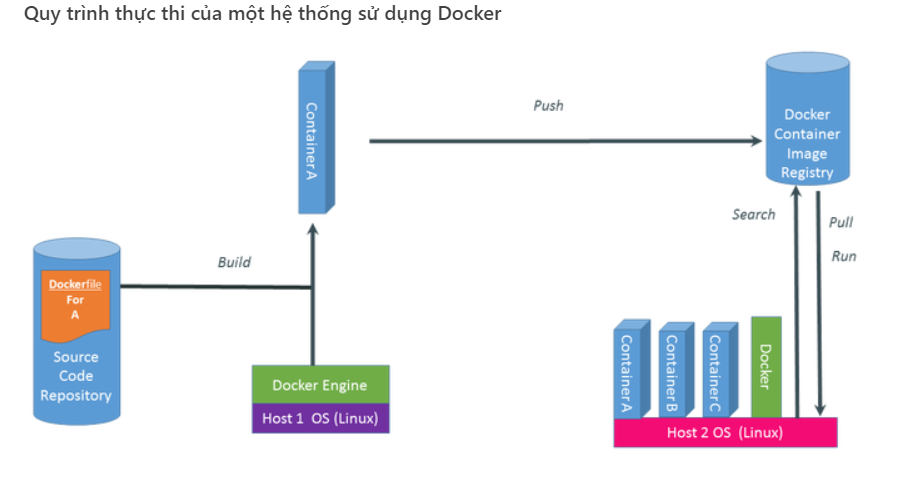
Sử dụng lệnh <docker images> để xem danh sách các images đã build

4.Run container của bạn:

Sử dụng lệnh sau để chạy container từ image đã build:

docker run <container\_name>

**Mô tả mô hình hoạt động**



V. Các lệnh cơ bản trong docker

List image/container:

$ docker image/container ls

Delete image/container:

$ docker image/container rm <tên image/container >

Delete all image hiện có:

$ docker image rm $(docker images –a –q)

List all container hiện có:

$ docker ps –a

Stop a container cụ thể:

$ docker stop <tên container>

Run container từ image và thay đổi tên container:

$ docker run –name <tên container> <tên image>

Stop all container:

docker stop $(docker ps –a –q)

Delete all container hiện có:

docker image rm $(docker images –a –q)

Show log a container:

$ docker logs <tên container>

Build một image từ container:

$ docker build -t <tên container>

Tạo một container chạy ngầm:

$ docker run -d <tên image>

Tải một image trên docker hub:

$ docker pull <tên image>

Start một container:

$ docker start <tên container>