Diagram

Description automatically generated with medium confidence

Cần các Images để tạo ra container

Images là không thể thay đổi

Thực thi của images là các container

* Ứng dụng chạy được là do build ra container , có thể ghi dữ liệu vào container

Graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Xóa images

Thực thi images và chạy container

Text

Description automatically generated

**Docker ps [-a]** -> Các container đang chạy

**Docker start [ID]** -> chạy container

Vào container -> **docker attach [conatiner ID]**

Muốn thoát terminal của container nhưng vẫn muốn container chạy mà không dừng thì không dùng ‘exit’ mà nhấn

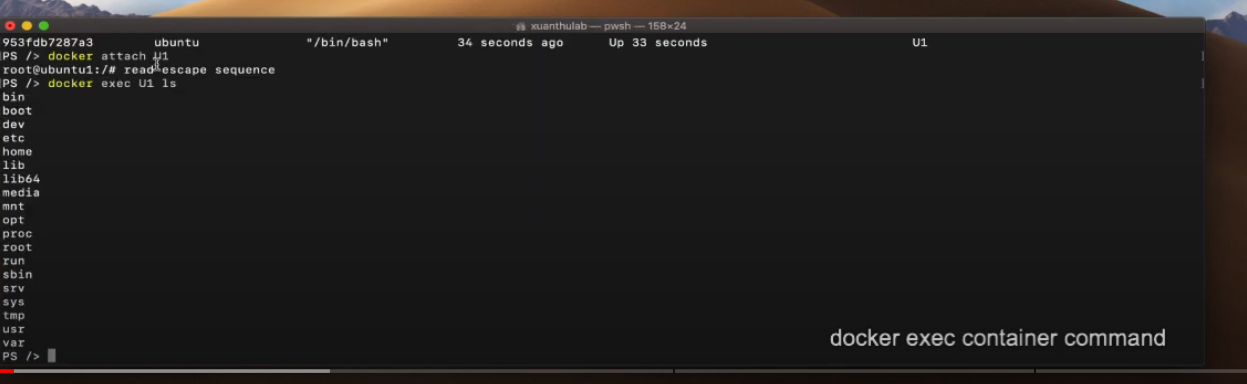
Graphical user interface, application

Description automatically generated

Dừng container đang chạy dùng **‘docker stop [id]’**

**Graphical user interface, application

Description automatically generated**

****

**Bên ngoài container nhưng muốn thi hành một lệnh của container đang chạy dùng lệnh**

**Docker exec [container] [lệnh]**

**Graphical user interface

Description automatically generated with medium confidence**

**Xuất image ra file**

**Text

Description automatically generated**

**Nạp image từ file vào**

**A picture containing graphical user interface

Description automatically generated**

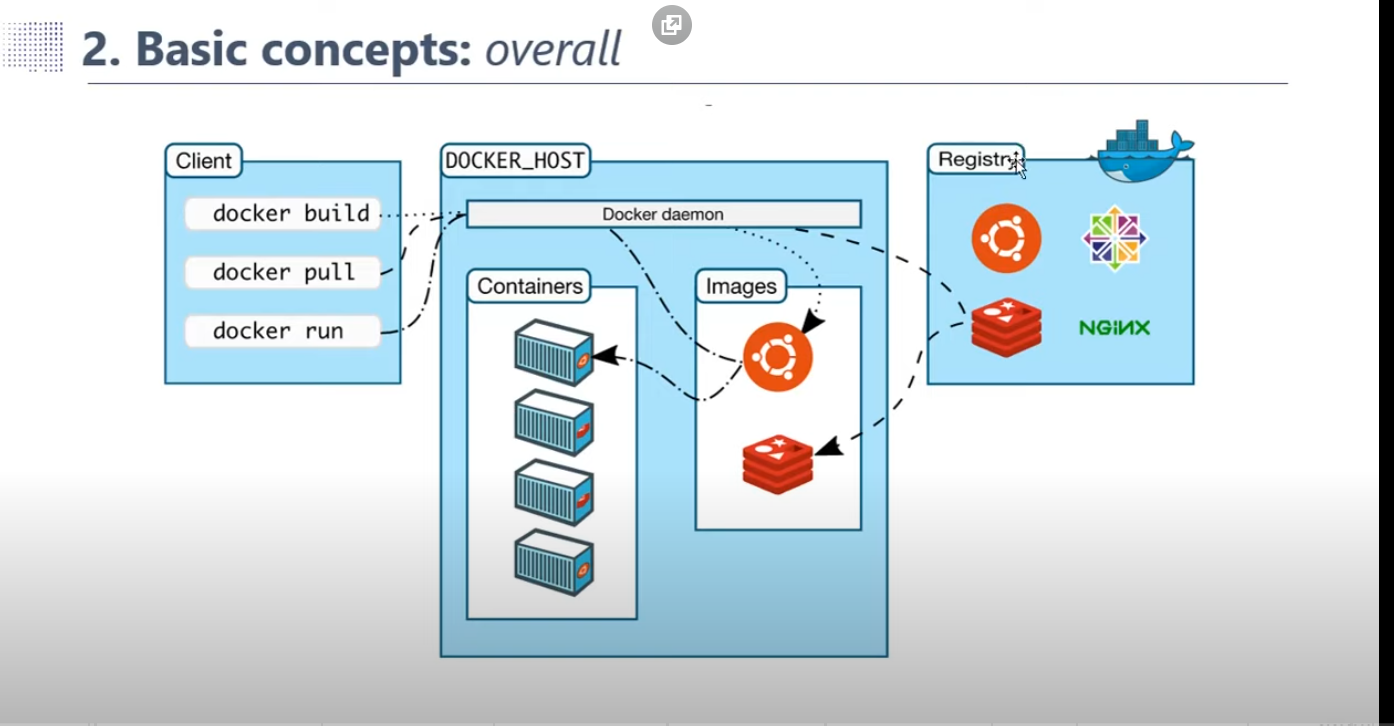
**Text

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Container sử dụng chung Kernel của máy host => Docker deamon phụ thuộc và kernel => gây ra một vài hạn chế không thể cài một số hệ điều hành lên nếu Kernel là window => Docker deamon sẽ được cài vào máy ảo linux để giải quyết hạn chế một số hệ điều hành không thể cài trên kernel của window**

****

**Docker CLI tương tác với docker daemon**

**Các images được lưu trữ trên Registry ( nơi lưu trữ các Images như Docker Hub )**

**Docker image ls === docker images**

**Text

Description automatically generated with low confidence**

**Thêm flag -a nếu muốn hiện luôn cả container đã shutdown**

**Do khi chạy container shell không có user -> bị exit(0) -> gắn flag ‘i’ và flag ‘t’**

* **Shell terminal trong container sẽ có user là chúng ta**
* **Graphical user interface, text, application

  Description automatically generated**
* **Khi chạy container nào cần có sự tương tác của người dùng**

**Dùng CTRL+D để thoát => Tiến trình PID = 1 bị tắt => container tắt luôn**

**Dùng CTRL+P+Q => Thoát terminal nhưng không tắt container**

**nginX là 1 loại Reverse proxy === Load balancer**

**Text

Description automatically generated**

**Dùng terminal SSH vào container để lấy nội dung HTML.**

**Thực tế không ai cho User SSH vào container**

**Cho nên ta sẽ mở PORT 80 của máy HOST và kết nối vào PORT 80 của Container để người dùng có thể truy cập nội dung**

**Flag -p viết tắt của (port)**

**Flag -d = Detact => chạy ngầm**

**Graphical user interface, text, application

Description automatically generated**

**Target\_port : Máy local**

**Graphical user interface, text, application, email

Description automatically generated**

**Ghi lại những thay đổi? Trong quá trình sử dụng container sẽ có thể sinh ra nhiều thay đổi, tạo mới, xóa nhiều file, thư mục… => Việc sử dụng bind mount để kết nối với thư mục trên máy host là không khả thi vì phải kiểm soát được thư mục nào thay đổi, file nào thay đổi…**

* **Tự tạo image riêng của mình.**

**Graphical user interface, text, application

Description automatically generated**

**Dấu . ở cuối là ‘Build context’ đây là tham số quan trọng trỏ đến thư mục đang chứa Dockerfile, bao gồm toàn bộ file, thư mục và ngay chính Dockerfile**

**Graphical user interface, text, application

Description automatically generated**

**Docker client sẽ gửi build context tới Docker Daemon để build image => tránh gửi quá nhiều nội dung**

**Trong quá trình build Image**

**Mỗi command trong DockerFile sẽ tạo 1 layer tạm (trong image có thể chứa nhiều image con)…**

**Sau khi làm xong thì xóa layer tạm đi**

* **Có thể giảm số lượng layer tạm bằng cách giảm số lượng command trong Dockerfile**

**Docker compose -> Triển khai nhiều container cùng lúc**

**Graphical user interface, application

Description automatically generated**

**Giao tiếp giữa các container**

**Tạo ra một đối tượng Network và gán các Container và trong đối tượng này**

* **Chúng có thể giao tiếp với nhau thông qua đối tượng Network đó thông qua tên Service**