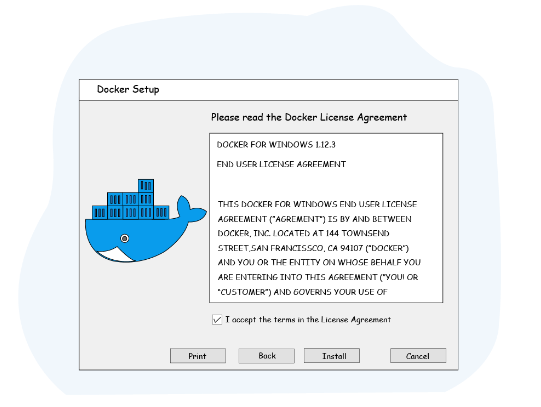
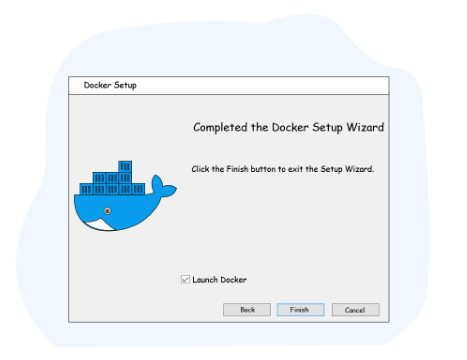
Nguyễn Văn Luận   
  
**Tìm hiểu về Docker**   
  
https://docs.docker.com/docker-for-windows/install/

Ta truy cập vào link rồi tải về và cài đặt Docker

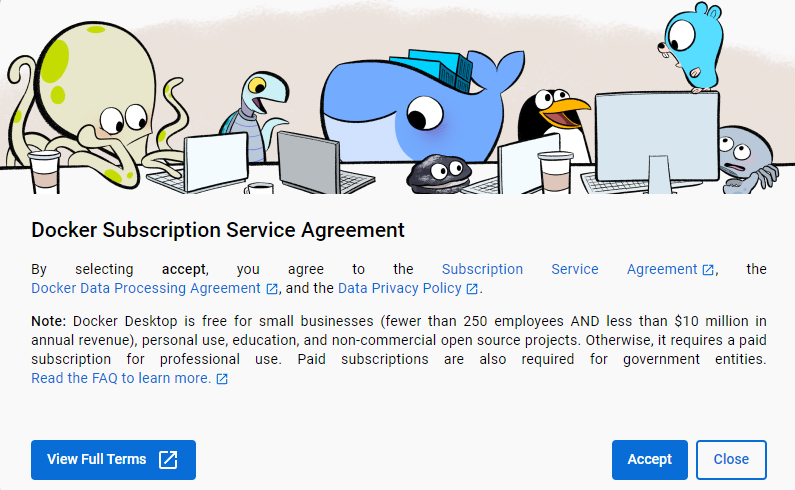
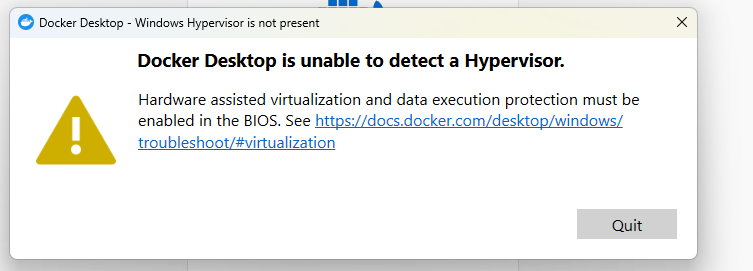
#### Bắt đầu bằng cách nhấp đúp vào trình cài đặt và sau đó tiếp tục với các bước sau:

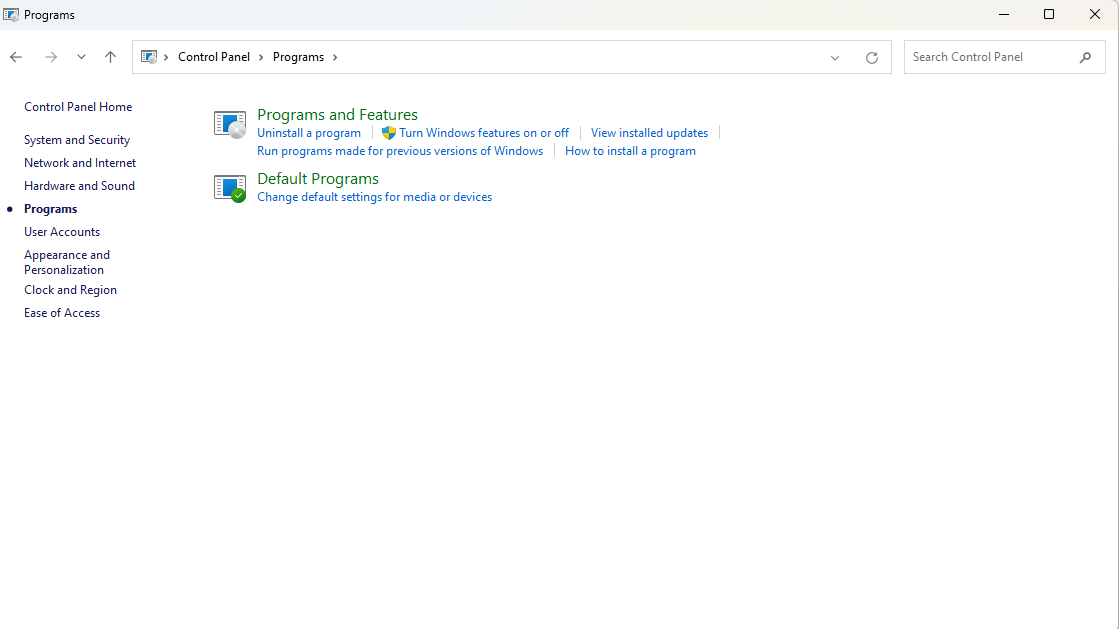
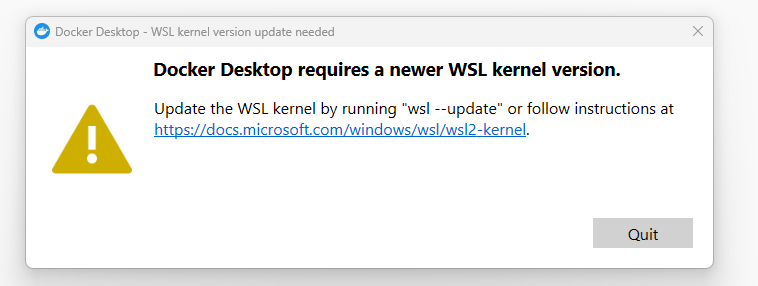
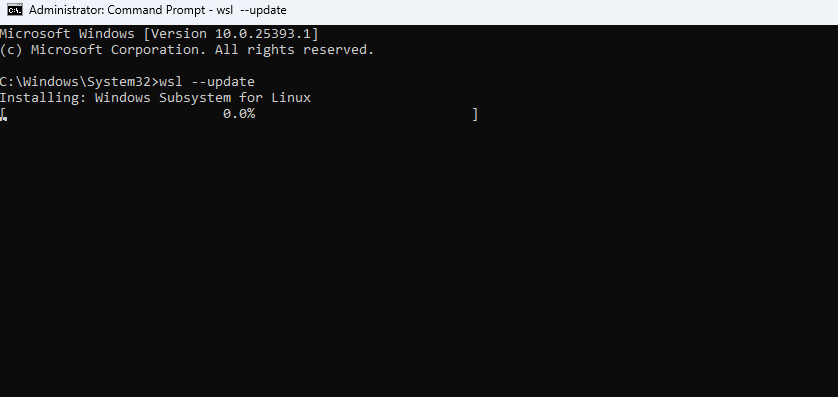
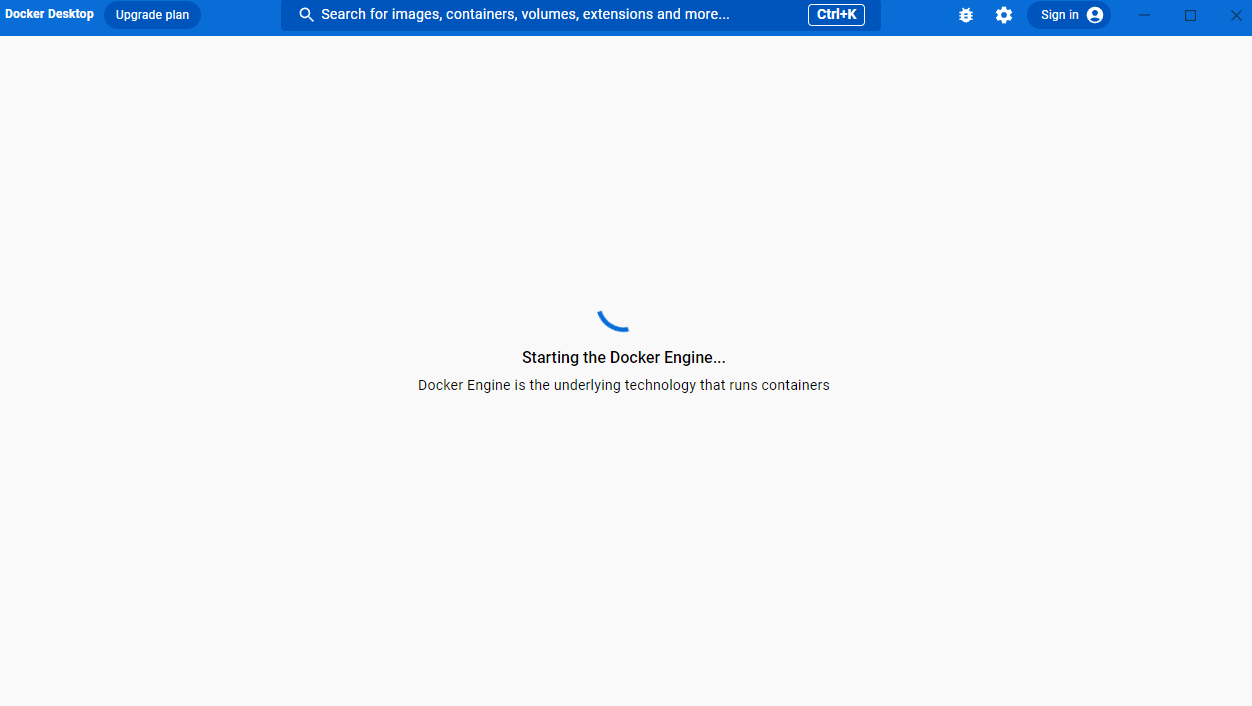
#### **Bước 1** : Đồng ý với các điều khoản là phải và tiếp tục với nút cài đặt

  
  
**Bước 2** : Để hoàn tất quá trình cài đặt chúng ta cần nhấn vào Finish



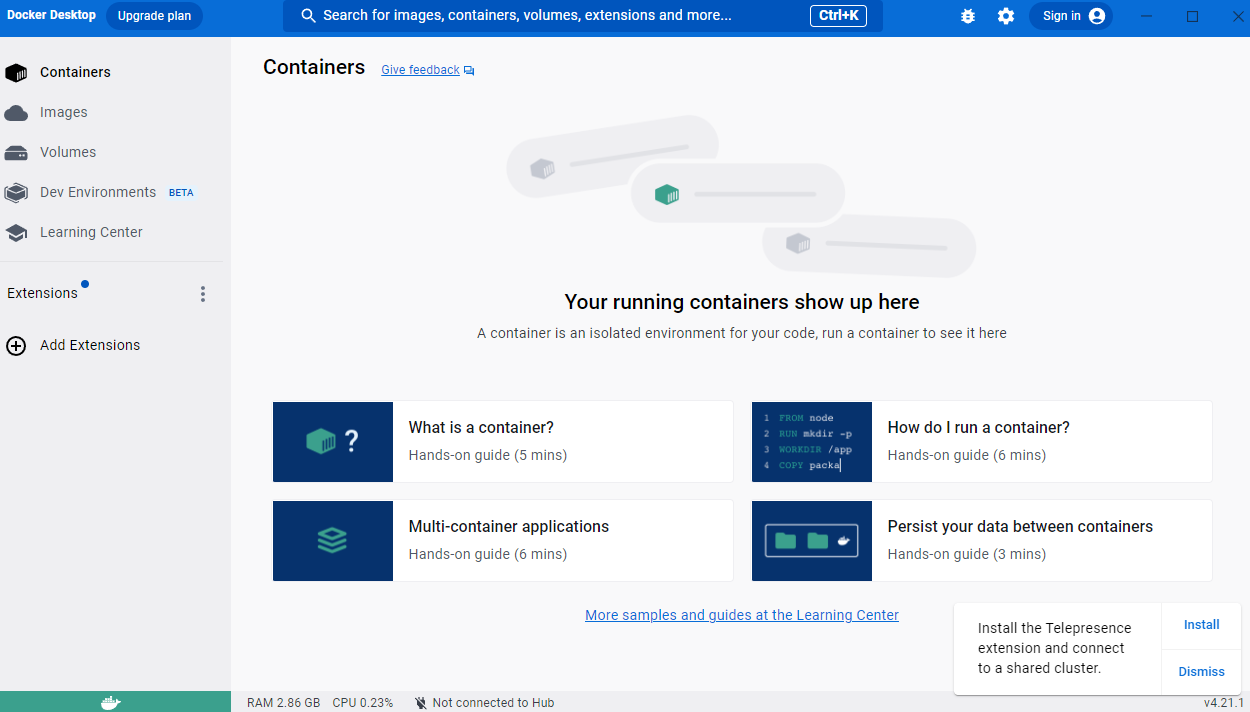
#### Sau khi cài đặt xong, nhấp vào nút bắt đầu và nhập Docker App.

  
  
  
  
Accept để tiếp tục:  
  
Nếu gặp phải lỗi này  
  
  
Ta sẽ xử lý như sau:  
  
Ta mở control panel -> Programs > Turn Windows features on off >

  
  
Rồi tìm đến   
  
  
rồi tích cho nó và Reset lại máy  
  
Trường hợp 2 nếu ta gặp lỗi này:  
  
  
  
Tiếp tục xử lý:  
  
  
Ta mở cmd rồi chạy lệnh  
wsl --update  
  
sau khi chạy xong câu lệnh trên ta mở Docker và được kết quả như sau:  
  


#### Nhấp đúp và mở Docker Hub sẽ cho bạn biết trạng thái của tất cả các hình ảnh và vùng chứa đã được tạo cho đến ngày hôm nay.

#### Tuy nhiên, đây là lần đầu tiên chúng ta chạy ứng dụng docker trong hệ thống của mình, nó sẽ trống.



Chúng ta sẽ thử chạy một cái gì đó trên Command Prompt để kiểm tra xem Docker có chạy đúng cách hay không.

Để kiểm tra xem Docker có chạy đúng cách hay không trên Command Prompt, bạn có thể thử chạy một vài lệnh đơn giản sau:

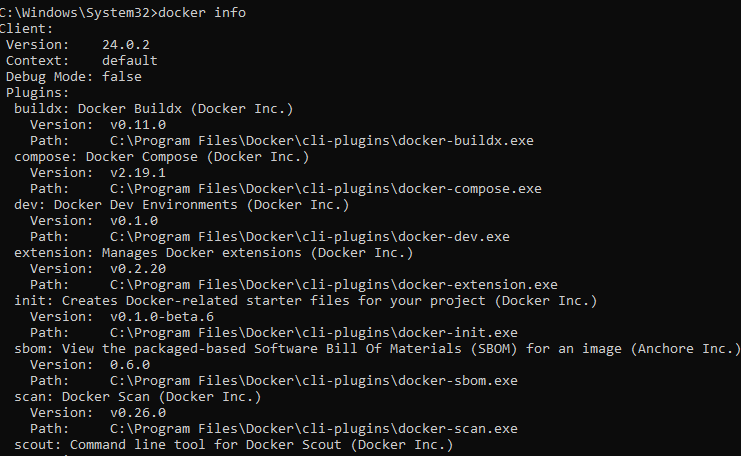
Kiểm tra phiên bản Docker:

docker --version



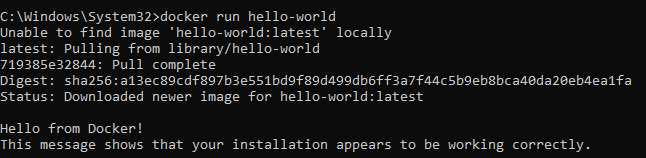
Kiểm tra thông tin Docker:

docker info



Chạy một container đơn giản:

docker run hello-world



Nếu ta nhìn thấy thông báo chào mừng và không có bất kỳ lỗi nào, có nghĩa là Docker đang hoạt động bình thường.

Cài đặt Docker Toolbox

#### Trước tiên hãy hiểu tại sao Docker Toolbox lại cần thiết.

#### Đối với người dùng phiên bản Windows 7 trở lên, Docker cung cấp cho chúng ta hộp công cụ Docker.

#### Nó là một trình cài đặt có mọi thứ cần thiết để định cấu hình và khởi chạy môi trường cho Docker.

#### Các hệ điều hành thiếu tài nguyên cần thiết cho ứng dụng docker mới, hộp công cụ sẽ xuất hiện và giúp chúng ta triển khai các vùng chứa phát triển.

#### Hộp công cụ Docker được phát triển cho Windows và MacOS để dễ dàng chạy các bộ chứa Docker.

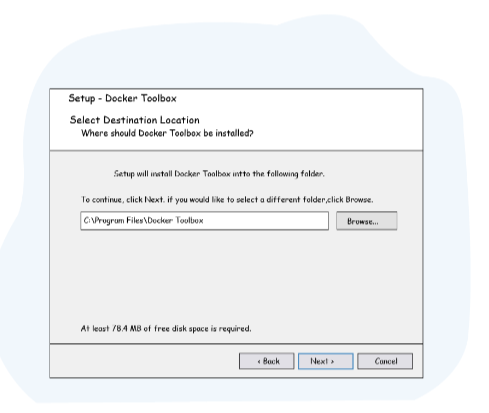
Link tải: [Releases · docker-archive/toolbox (github.com)](https://github.com/docker-archive/toolbox/releases)

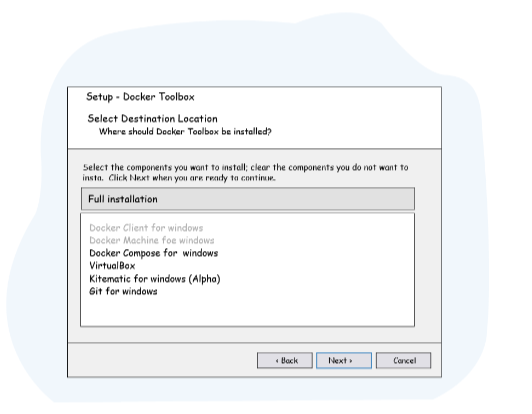
#### Khi bạn đã hoàn tất việc Tải xuống trình cài đặt, hãy nhấp đúp để bắt đầu với các bước sau:

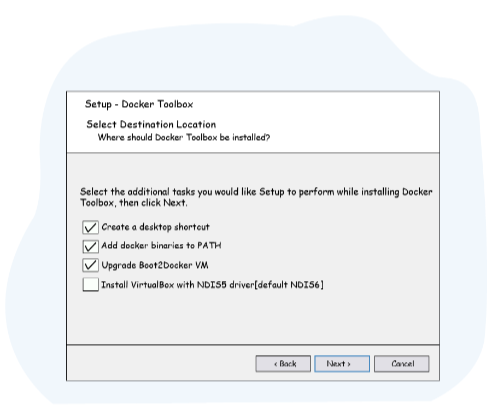
#### **Bước 1** : Màn hình đầu tiên sẽ có nút Next. Nhấp vào nó.



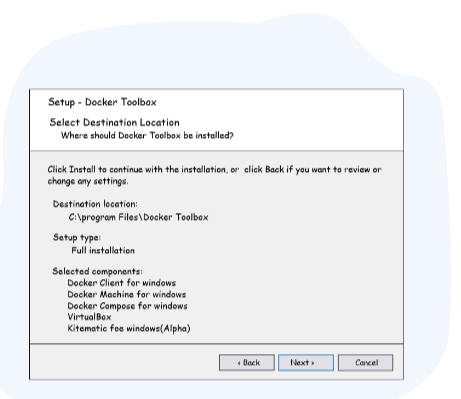
#### **Bước 2** : Nhập vị trí mặc định mong muốn của bạn và sau đó nhấp vào tiếp theo.

  
**Bước 3** : Để nguyên các thành phần mặc định và tiến hành Next

  
**Bước 4** : Hãy bắt đầu với việc giữ nguyên các tác vụ bổ sung và nhấn Next.



#### **Bước 5** : Cuối cùng là Click Install



Và bạn đã hoàn tất việc cài đặt Hộp công cụ Docker.

Làm tốt lắm, chúng ta chuẩn bị lên đường!

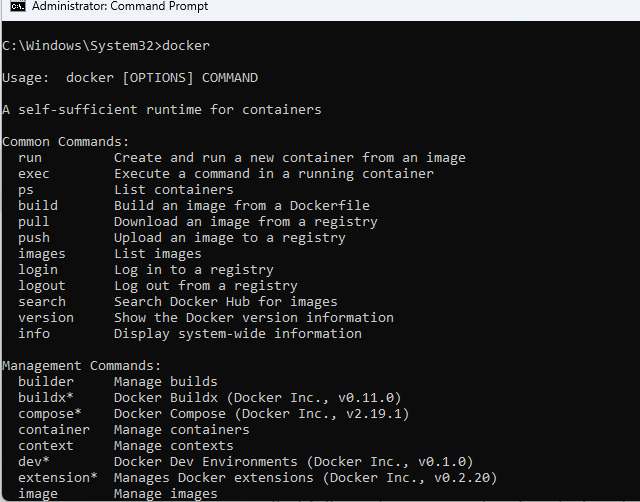
### **Chúng ta sẽ tìm hiểu về Docker commands**

#### Chúng ta hãy xem từng lệnh docker.

#### Chúng ta sẽ chạy các lệnh này trên dòng lệnh của mình như chúng ta đã thực hiện trong chương trước.

#### Nếu phải tìm danh sách các lệnh có sẵn, hãy chạy docker lệnh không có tham số. Điều này sẽ in danh sách các lệnh có sẵn.

$ docker

  
  
Cú pháp chung sẽ là:

docker [OPTIONS] COMMAND [ARG...]

docker images

Lệnh hình ảnh docker liệt kê tất cả các Hình ảnh Docker có trên Máy chủ Docker của bạn.



  
  
Đầu ra của lệnh này đại diện cho tên của hình ảnh docker, thẻ của nó, Id hình ảnh, thời gian tạo và kích thước tương ứng.

### 

### **Một số lệnh khác**

docker rmi

#### Lệnh docker rmi cho phép chúng ta xóa một hoặc nhiều hình ảnh khỏi máy chủ docker.

docker exec

#### Lệnh exec được sử dụng để chạy một lệnh mới trong vùng chứa đang chạy.

#### Lệnh bắt đầu sử dụng docker exec chỉ chạy trong khi quy trình chính của bộ chứa đang chạy và nó không được khởi động lại nếu bộ chứa được khởi động lại.

docker history

#### Lệnh này hiển thị lịch sử của hình ảnh docker với tên hình ảnh được đề cập trong lệnh.

docker info

#### Lệnh này giúp hiển thị Thông tin hệ thống.

#### Thông tin chi tiết về docker được cài đặt trên hệ thống bao gồm phiên bản kernel, số lượng bộ chứa và hình ảnh, v.v.

docker pull

#### Lệnh này lấy một hình ảnh từ sổ đăng ký

#### Theo mặc định, docker pull kéo một hình ảnh từ sổ đăng ký. Một kho lưu trữ có thể chứa nhiều hình ảnh.

### **Container Commands**

#### 

#### Mọi người, bây giờ chúng ta hãy xem các lệnh giúp chúng ta làm việc với Docker container.

#### Bắt đầu từ liệt kê chúng để đổi tên chúng và nhiều hơn nữa. Cứ đi đi.

  
docker ps

#### Đây là lệnh cần thiết để liệt kê các bộ chứa docker hiện có ở trạng thái đang chạy. Ở đây, ps là viết tắt của “Process Status”.

#### Lệnh **ps** được sử dụng để mô tả trạng thái của tiến trình.

docker create

#### Lệnh này giúp tạo một vùng chứa mới là lớp có thể ghi trên hình ảnh đã chỉ định. Nó cũng chuẩn bị để chạy một lệnh cụ thể.

docker attach

#### Lệnh này sẽ đính kèm các luồng đầu ra, đầu vào và lỗi tiêu chuẩn của thiết bị đầu cuối của bạn vào một bộ chứa đang chạy bằng ID hoặc tên của nó.

#### Điều này cho phép bạn xem đầu ra đang diễn ra hoặc kiểm soát hoạt động của nó, như thể các lệnh đang chạy trực tiếp trong thiết bị đầu cuối của bạn.

docker stop

#### Lệnh dừng docker dừng container một cách duyên dáng, đó là sự khác biệt giữa lệnh kill và lệnh dừng.

docker kill

#### Lệnh này giúp tắt một hoặc nhiều vùng chứa đang chạy.

docker rename

#### Lệnh này giúp đổi tên vùng chứa.

#### Khi bộ chứa Docker được tạo, hệ thống sẽ tự động gán số nhận dạng duy nhất toàn cầu (UUID) cho mỗi bộ chứa để tránh mọi xung đột đặt tên và cải thiện khả năng tự động hóa mà không cần sự tham gia của con người.

docker update

#### Lệnh này được sử dụng để cập nhật cấu hình của một hoặc nhiều vùng chứa Lệnh cập nhật docker tự động cập nhật cấu hình vùng chứa.

#### Chúng ta có thể sử dụng lệnh này để ngăn các container tiêu thụ quá nhiều tài nguyên từ máy chủ Docker của chúng.

#### 

#### **Tổng quan Docker Image**



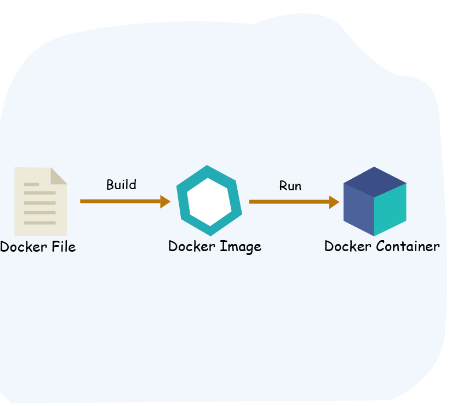
#### Hãy tưởng tượng nếu bạn cần xây dựng một ngôi nhà cho chính mình. Hãy cùng nắm bắt quy trình chúng ta cần làm để xây dựng một ngôi nhà.

Đầu tiên, chúng ta cần tạo bản thiết kế của nó với tất cả thông tin chi tiết của từng phòng với kích thước, cách cấu trúc của nó, v.v.

#### Tương tự, trong Docker, để xây dựng một môi trường như vậy, chúng ta cần tạo bản thiết kế.

#### Kế hoạch chi tiết này có dạng DockerFile

#### Bản thiết kế là tập hợp các lệnh theo cách có trật tự để thiết lập môi trường và sẽ chạy theo mặc định khi khởi động.



#### Theo cách tương tự, hình ảnh Docker là bản thiết kế của các thùng chứa, chứa thông tin bắt buộc mà chúng ta cần để một thùng chứa lắp ráp.

#### Một vùng chứa được tạo bằng cách chạy một hình ảnh.

#### Hình ảnh Docker là một tệp, bao gồm nhiều lớp, được sử dụng để thực thi mã trong vùng chứa.

#### Như đã thảo luận trước đó, hình ảnh được xây dựng từ một tập hợp các hướng dẫn để hoàn thành và xây dựng phiên bản thực thi của ứng dụng, dựa trên nhân hệ điều hành máy chủ.

#### Sau khi người dùng Docker chạy một hình ảnh, nó sẽ trở thành một hoặc nhiều phiên bản của vùng chứa đó.

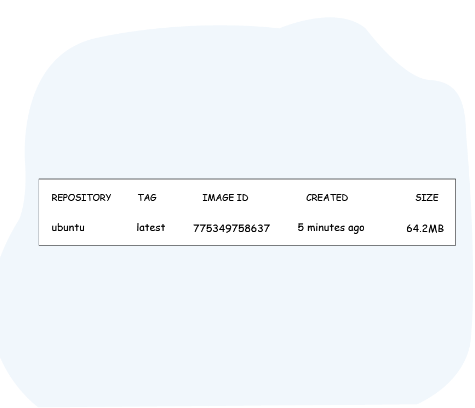
#### Hình ảnh Docker bao gồm tất cả các yếu tố cần thiết để chạy ứng dụng dưới dạng vùng chứa. Chúng bao gồm mã, tệp cấu hình, biến môi trường, thư viện và thời gian chạy.

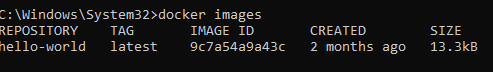
#### Lệnh run tạo một vùng chứa từ hình ảnh đã cho. Hình ảnh Docker cũng là một tài sản có thể tái sử dụng có thể được triển khai trên bất kỳ máy chủ nào.

#### Để biết danh sách các Docker image trong hệ thống của bạn, chúng ta sử dụng lệnh images.

$ docker images

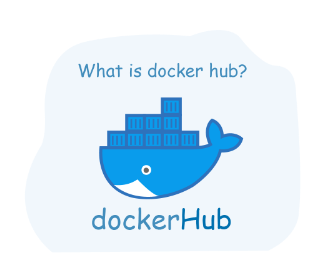
#### Chúng ta nhận được một loại đầu ra tương tự như dưới đây:





#### Nó cũng trả về kho lưu trữ, thẻ, Id hình ảnh, thời gian tạo và kích thước của tệp. Đây là tất cả về hình ảnh docker. Hãy di chuyển xa hơn.

### **What is Docker Hub?**

  
Một kho lưu trữ tất cả các docker image trên cloud, có thể nói là Docker Hub.

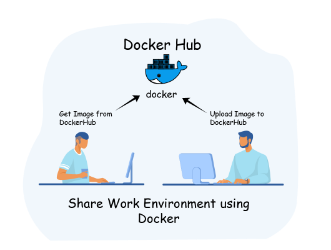
#### Chứa môi trường và xây dựng hình ảnh là công việc thực tế của Docker.

#### Và sau đó hình ảnh có thể được đẩy vào Docker Hub.

#### Hình ảnh được lưu trữ trong Docker Image Registry cho mục đích tổ chức và cộng tác. Đó là Docker Hub.

#### Để chạy bất kỳ công cụ nào, một hình ảnh được lấy từ công cụ đó từ Hub và chạy dưới dạng vùng chứa.

#### Nó cũng giúp bạn tạo một môi trường tự động trong đó các thay đổi được thực hiện bởi nhà phát triển có thể được xây dựng, thử nghiệm và đẩy, ngoài việc lưu và vận chuyển các thùng chứa.

  
Ví dụ: Chúng ta có thể kéo các bộ chứa windows hoặc thậm chí các bộ chứa ubuntu và chạy.

#### Từ Docker Hub, các thùng chứa này được tải xuống hệ thống của bạn dưới dạng hình ảnh. Chúng ta có thể làm bất cứ điều gì với những hình ảnh này bằng cách sử dụng các lệnh.

#### Như chạy, tạm dừng, dừng hoặc thậm chí xóa hình ảnh vùng chứa.

#### Docker Hub cũng có thể nói là nơi chúng ta có thể tìm thấy bất kỳ hình ảnh docker nào có sẵn để sử dụng.

#### Docker Images mà chúng ta cần để sử dụng có thể được tìm thấy trong kho lưu trữ trực tuyến chính thức có tên là Docker Hub.

#### Nói thêm về trung tâm docker, nó cũng cung cấp cho chúng ta cơ sở để lưu trữ và phân phối hình ảnh tùy chỉnh của chúng ta nếu chúng ta muốn làm như vậy.

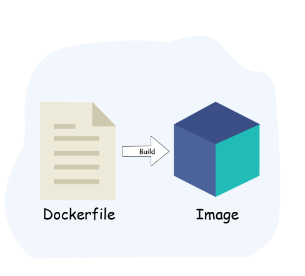
#### Dựa trên các yêu cầu của chúng ta, chúng có thể được đặt ở chế độ công khai hoặc riêng tư.

### **Tóm tắt lại**



* Hình ảnh Docker là bản thiết kế của các thùng chứa, chứa thông tin bắt buộc mà chúng ta cần để một thùng chứa lắp ráp.
* Một kho lưu trữ tất cả các docker image trên cloud, có thể nói là Docker Hub.

### **Tạo Docker File**



#### Docker cung cấp cho chúng ta các tính năng bổ sung và một trong số đó là Docker File.

#### Docker cung cấp cho chúng ta chức năng để tạo hình ảnh Docker của riêng chúng ta.

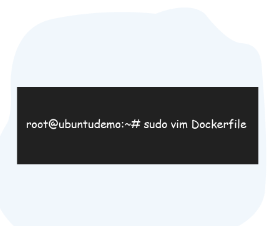
#### Điều này được thực hiện với sự trợ giúp của Docker Files.

#### Tệp Docker có thể nói là một tệp văn bản đơn giản với một bộ hướng dẫn để xây dựng hình ảnh của chúng ta.

#### Hãy để chúng ta xem cách chúng ta có thể tạo tệp Docker từng bước -

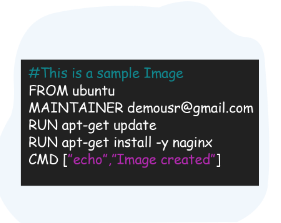
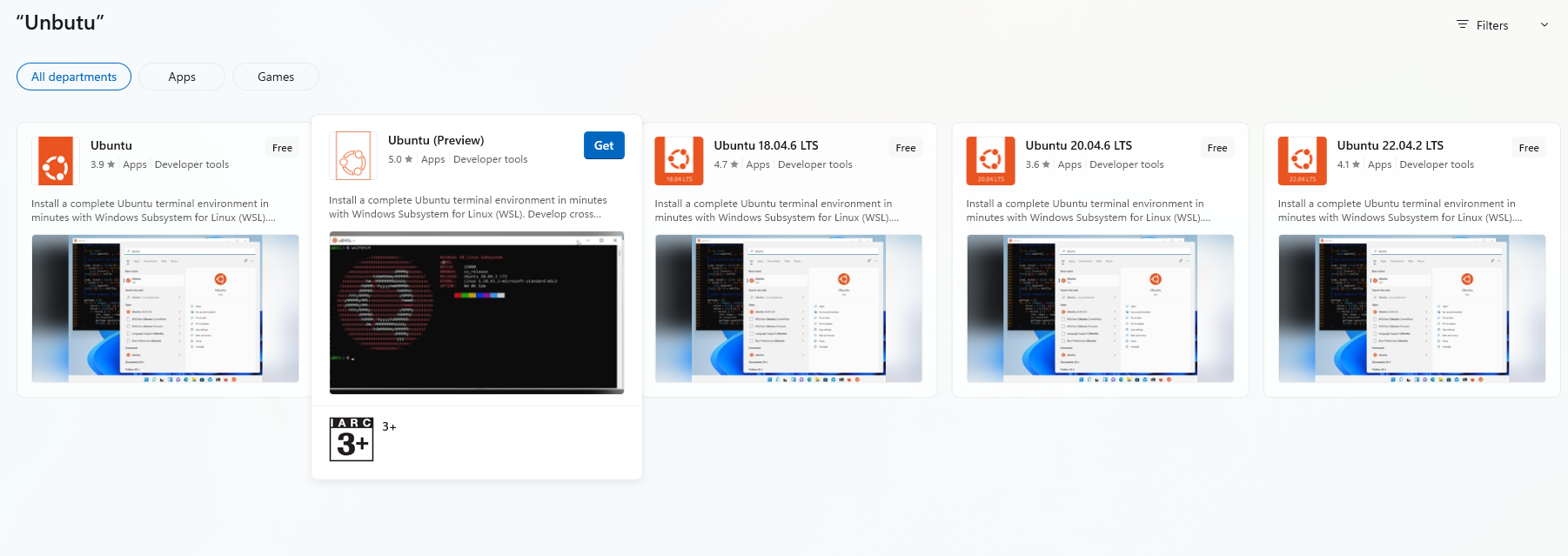
#### Chúng ta cần tạo một Docker File và chỉnh sửa nó bằng vim.

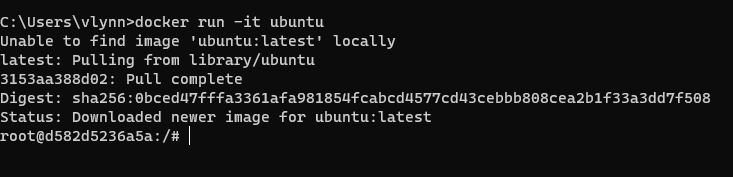
#### **Lưu ý** : Tên của tệp phải là Dockerfile với D là VỐN viết hoa chữ cái đấu

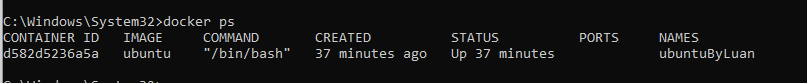
  
  
Vì vim không có sẵn trên Window nên ta có thể thay thế bằng notepad hoặc notepad++ để viết:   
  
  
  
Chúng ta cần lưu ý để tạo DockerFile:

* Dòng đầu tiên bao gồm các nhận xét, bắt đầu bằng '#'
* Dòng tiếp theo phải bắt đầu bằng từ khóa FROM. Nó giúp chỉ định nơi chúng ta cần tạo
* Hình ảnh cơ sở cho hình ảnh.
* Lệnh tiếp theo bao gồm người sẽ duy trì hình ảnh. Ở đây chúng ta cần chỉ định từ khóa MAINTAINER và id email của họ.
* Lệnh RUN sẽ chạy các hướng dẫn đối với hình ảnh.
* Lệnh cuối cùng sẽ chỉ định thông báo chúng ta cần hiển thị cho Người dùng.

#### Sau khi hoàn thành, chúng ta cần lưu tệp này để có thể tạo hình ảnh từ Tệp Docker này.

  
  
Để chạy câu lệnh kiểu này, chúng ta phải cài hệ điều hành Unbuntu cho nó, chúng ta có thể cài trực tiếp trên microsoft store hoặc chạy bằng lệnh để nó chạy trong một container của Docker   
  
  
  
hoặc docker run -it ubuntu

  
  
sau khi thông báo Pull complete nghĩa là nó đã thành công

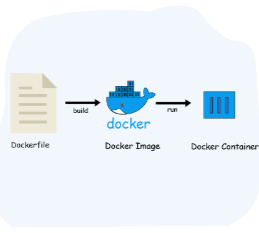
sau đó ta chạy lệnh docker ps để xem các container trong Docker của ta  
  
  
  
ở đây ta có thể chạy thực thi chạy ubuntu thông qua lệnh

thông qua Id của container

docker exec -it d582d5236a5a bash

hoặc thông qua names   
docker exec -it ubuntuByLuan bash

### **Lệnh DockerFile**



#### Trong chương trước, chúng ta đã tạo Tệp Docker. Bây giờ, chúng ta sẽ xem cách xây dựng Docker File.

#### Tệp có thể được tạo bằng cách sử dụng lệnh “build”.

#### Lệnh này giúp chúng ta xây dựng hình ảnh docker từ DockerFile.

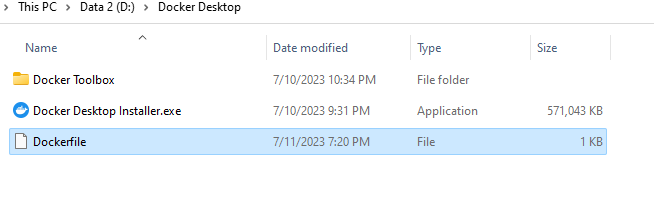
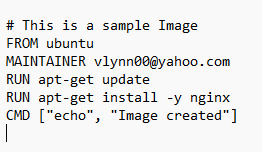
#### Cú pháp cho nó diễn ra như sau,

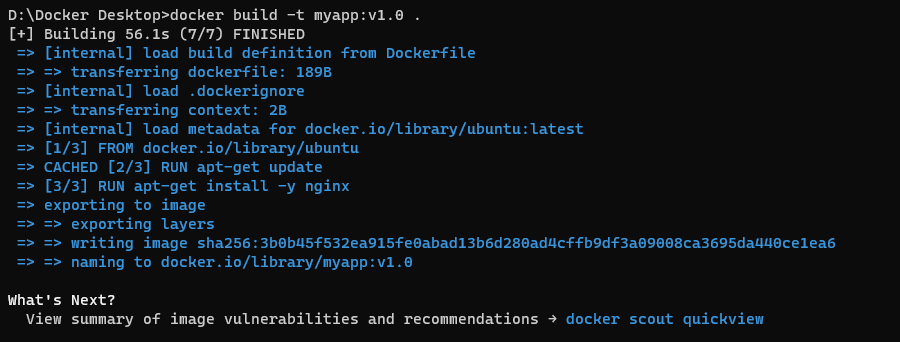
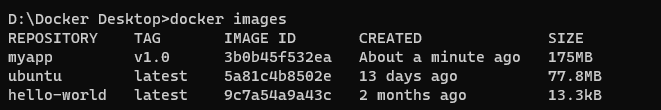
docker build -t ImageName: TagName dir

#### Đây,

* -t − để đề cập đến một thẻ cho hình ảnh
* ImageName − Tên mà chúng ta muốn gán cho hình ảnh của mình.
* TagName − thẻ chúng tôi muốn cung cấp cho hình ảnh của mình.
* Dir − Thư mục của Docker File.

docker build -t myapp:v1.0 .

ta chạy cmd trong thư mục đường dẫn tới DockerFile  
  
  
file docker ta chuẩn bị trước đó có nội dung như sau:  
  
ta nhớ đổi xóa định dạng đuôi file .txt để trở thành file gốc để có thể chạy được   
  
rồi chạy lệnh sau:  
docker build -t myapp:v1.0 .

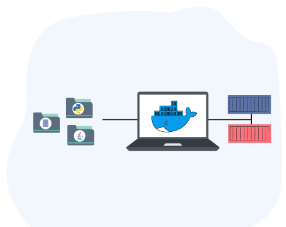
và được kết quả như sau:  
  
  
sau đó ta chạy lệnh docker images thì đã thấy nó được cài đặt   


### **Tóm tắt.**

* DockerFile là một tài liệu văn bản với một bộ hướng dẫn được sử dụng để xây dựng
* Docker Image
* Docker Image lần lượt được sử dụng để xây dựng Docker Container
* Tệp Docker được sử dụng để xây dựng tệp hình ảnh tùy chỉnh của chúng ta.

### **Docker Architecture**

### **cơ chế**



#### Docker Engine là thành phần chính của toàn bộ hệ thống Docker.

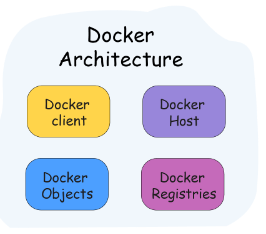
#### Docker tuân theo kiến ​​trúc máy khách-máy chủ.

#### Nó có ba phần chính,

* Docker Client
* Docker Host
* Registry

#### Docker Daemon đóng vai trò chính là xây dựng, chạy và phân phối các Docker container.

#### Ứng dụng khách Docker tương tác với daemon Docker chạy trên cùng một hệ thống. Chúng kết nối với nhau thông qua một giao diện mạng.



#### Bây giờ chúng ta hãy hiểu từng cái một.

### **Các thành phần**

#### Docker daemon

#### Còn được gọi là Dockerd. Nó có thể tạo và quản lý các đối tượng bao gồm Vùng chứa, hình ảnh và mạng, v.v.

#### Rest API hướng dẫn nó những gì phải được thực hiện.

#### Và dòng lệnh (cli) là nơi khách hàng tương tác bằng cách sử dụng các lệnh.

### **Docker Client**

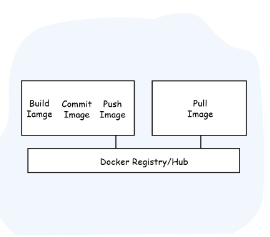


#### Người dùng tương tác với Docker thông qua Docker Client.

#### Khi các lệnh docker được chạy, nó sẽ được gửi qua Máy khách đến Daemon. Lệnh Docker sử dụng API Docker.

#### Máy khách có thể tương tác với nhiều Docker Daemon.

### **Docker Registries**



#### Vị trí lưu trữ hình ảnh Docker là docker Registry.

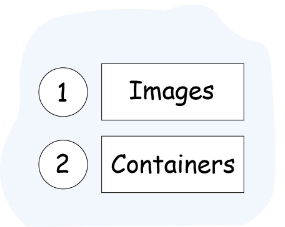
#### Vị trí mặc định của Image là Docker Hub, có thể nói là Public registry.

#### Người dùng cũng có tùy chọn tạo Sổ đăng ký riêng của họ.

#### Các lệnh Docker như Docker Pull và Docker Run, lấy hình ảnh từ sổ đăng ký đã định cấu hình.

#### Trong khi lệnh Docker Push, hình ảnh được lưu trữ trong Sổ đăng ký được định cấu hình.

### **Docker Objects**



#### Như đã thảo luận trước đó, Docker Images, Container, mạng đều là các Đối tượng Docker.

#### Hãy có một cái nhìn tổng quan về nó.

### **Docker Images**

#### Mẫu chỉ đọc với Bộ hướng dẫn[Template] để tạo Bộ chứa Docker được gọi là Hình ảnh Docker.

#### Docker Hub cung cấp cho chúng ta Hình ảnh, Ngoài ra, chúng ta cũng có thể tùy chỉnh hình ảnh bằng cách thêm các hướng dẫn bổ sung vào hình ảnh cơ sở.

#### Chúng ta có thể tạo hình ảnh Docker của riêng mình từ Docker Files. Để làm được điều đó, chúng ta cần tạo một dockerfile với bộ hướng dẫn.

#### Khi chúng ta chạy Dockerfile này, nó sẽ tạo hình ảnh tùy chỉnh của chúng ta, từ đó giúp tạo vùng chứa.

#### Trong Tệp Docker, Mỗi lệnh tạo một lớp trên hình ảnh.

#### Các thay đổi trong dockerfile, xây dựng lại hình ảnh. Nhưng chỉ những lớp đã thay đổi mới được xây dựng lại.

#### Tính năng quan trọng này giúp làm cho Docker Nhẹ và Nhanh hơn so với các Công nghệ khác.

### **Docker Containers**



#### Khi một hình ảnh được chạy, nó sẽ tạo một Docker Container.

#### Chúng ta có thể tạo, bắt đầu, dừng, di chuyển hoặc thậm chí xóa Vùng chứa bằng Dòng lệnh.

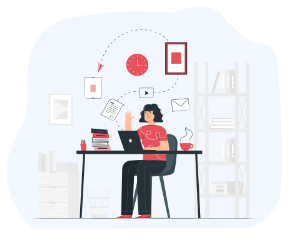
#### Các vùng chứa đã được cách ly với các vùng chứa khác và cả với Thiết bị chủ.

### **Ưu điểm của docker**

#### Docker Container hữu ích và tiện dụng hơn rất nhiều vì chúng cho phép chúng tôi linh hoạt xây dựng các ứng dụng khác nhau, có thể dễ dàng bảo trì và di chuyển trơn tru từ hệ thống này sang hệ thống khác mà không có bất kỳ tác động nào đến các đối tác thông thường của chúng.

#### Chúng ta hãy xem những lợi thế của Docker trong chương này.

### **Tính di động**

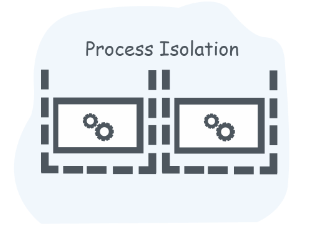


#### Bộ chứa docker có thể chạy trên bất kỳ máy nào hỗ trợ môi trường của bộ chứa.

#### Các ứng dụng không phụ thuộc vào hệ điều hành máy chủ, do đó cả hệ điều hành và môi trường ứng dụng đều có thể được giữ sạch sẽ.

#### Sự thay đổi này giúp dễ dàng chuyển các ứng dụng dựa trên vùng chứa sang môi trường đám mây hoặc từ máy tính xách tay của nhà phát triển sang máy chủ, miễn là hệ thống đích hỗ trợ Docker để thử nghiệm hoặc xử lý thêm.

### **Cô lập và điều tiết**



#### Các thùng chứa này không chỉ giữ cho hệ thống máy chủ cơ bản bị cô lập mà còn giữ cho các ứng dụng được cách ly với nhau.

#### Nó giúp các ứng dụng được đóng gói sử dụng tài nguyên hệ thống như CPU, Bộ nhớ, thiết bị I/O, mạng, v.v. dễ dàng hơn.

#### Ngoài ra, để tránh nhầm lẫn, dữ liệu và mã được giữ riêng biệt.

* Hiệu suất

#### Mọi người thường có một quan niệm sai lầm khi thực sự coi các container là máy ảo.

#### Nhưng cả hai đều hoạt động trên các cơ chế khác nhau và do đó có những ưu và nhược điểm khác nhau.

#### Các thùng chứa chỉ sử dụng các phần cần thiết của tài nguyên hệ điều hành.

#### Nhiều ứng dụng chia sẻ cùng một Nhân hệ điều hành theo cách được quản lý tốt.

#### Vì các vùng chứa không chứa Hệ điều hành nên chúng có dấu chân nhỏ hơn nhiều so với Vms.

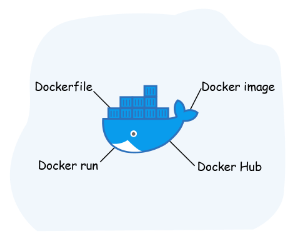
#### Do đó được tạo ra nhanh hơn và khởi động nhanh hơn nhiều, thay vì giảm nguy cơ ngừng hoạt động.

* khả năng mở rộng

#### Nếu ứng dụng yêu cầu, chúng ta có thể nhanh chóng tạo vùng chứa mới.

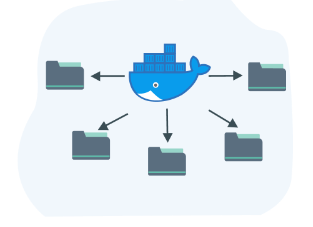
#### Tùy chọn quản lý vùng chứa cung cấp cho chúng ta lợi thế của việc sử dụng nhiều vùng chứa cùng một lúc.

### **Tóm tắt lại**



* Docker cải thiện Hiệu suất, Khả năng mở rộng, làm cho nó có khả năng truy cập cao, đồng thời cung cấp một môi trường biệt lập cho Ứng dụng.
* Docker sử dụng bộ nhớ và tài nguyên tối thiểu.
* Docker cũng làm giảm khả năng xảy ra xung đột. Và giảm thời gian ngừng hoạt động xảy ra khi chúng ta sử dụng VM.

### **Docker Storage**



#### Với thời gian Chuyển đổi, Docker là một trong những công nghệ được Chuyển đổi nhiều nhất trên thế giới ngày nay.

#### Nó đã ảnh hưởng đến nhiều phần khác nhau của thế giới CNTT bao gồm cả Lưu trữ.

#### Bộ lưu trữ Docker giúp định cấu hình và hỗ trợ lưu trữ dữ liệu ứng dụng trong quá trình triển khai vùng chứa.

#### Các chức năng vùng chứa khác được xử lý bởi các thư viện phần mềm bổ sung.

#### Các thư viện này bao gồm libnetwork và swarmkit tương ứng cho khả năng quản lý mạng và lưu trữ.

### **Storage Drivers**

#### Docker sử dụng nhiều loại trình điều khiển lưu trữ khác nhau cho mục đích lưu trữ. Tuy nhiên, các trình điều khiển lưu trữ khác nhau sử dụng các công nghệ khác nhau.

#### Trình điều khiển lưu trữ tạo dữ liệu trong lớp có thể ghi của Vùng chứa của chúng ta.

#### Khi Vùng chứa bị xóa, các tệp đó sẽ khả dụng.

#### Nhưng tốc độ đọc và ghi trở nên thấp hơn.

#### Hãy để chúng ta có một cái nhìn tại nó.

#### **AUFS:**

1. AUFS là một hệ thống tệp hợp nhất. Đây là trình điều khiển lưu trữ mặc định được sử dụng để quản lý hình ảnh và lớp trên Docker Đối với các phiên bản Ubuntu trước đây.
2. Trình điều khiển lưu trữ AUFS sử dụng công nghệ aufs và nó đảm bảo chức năng trơn tru của các bộ chứa với mức sử dụng bộ nhớ tốt.
3. Nó là một trình điều khiển ổn định, được sử dụng cho các ứng dụng sẵn sàng sản xuất.
4. Một Chức năng của AUFS được xem xét là Hoạt động ghi cao của nó.

#### **DEVICEMAPPER:**

1. Devicemapper cũng là một trình điều khiển Ổn định và cũng đảm bảo Trải nghiệm Docker mượt mà.
2. Nó giúp người thử nghiệm kiểm tra các ứng dụng trong phòng thí nghiệm.
3. Trình điều khiển này hoạt động ở cấp Khối hơn là cấp Tệp.

#### **Btrfs:**

1. Btrfs hỗ trợ các công nghệ lưu trữ tiên tiến giúp nó trở nên tuyệt vời cho docker và cũng là một hệ thống tệp sao chép khi ghi thế hệ tiếp theo.
2. Nó được bao gồm trong nhân Linux chính.
3. Nó cũng có hoạt động ghi cao.
4. Trình điều khiển này được sử dụng tối đa để quản lý hình ảnh và vùng chứa.

#### **Overlay:**

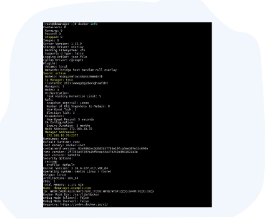
1. Overlay là một hệ thống tệp hợp nhất tương tự như AUFS, với khả năng triển khai nhanh hơn.
2. Nó là một trình điều khiển ổn định và sử dụng bộ nhớ tốt.
3. Nó là tốt cho các ứng dụng thử nghiệm trong phòng thí nghiệm

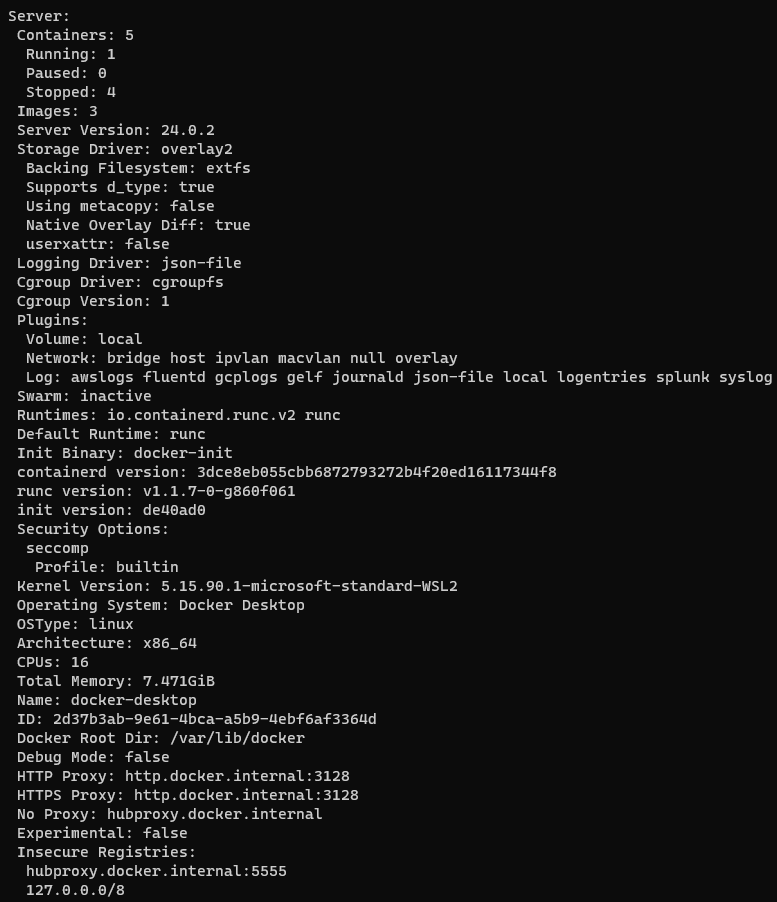
#### **ZFS:**

1. ZFS hỗ trợ các công nghệ lưu trữ tiên tiến như nén, chống trùng lặp, quản lý ổ đĩa, ảnh chụp nhanh, sao chép, v.v.
2. Nó cũng là một trình điều khiển ổn định và tốt cho các ứng dụng thử nghiệm
3. Nó được phát triển bởi Sun Microsystems.

### **Docker command**

#### Lệnh docker để kiểm tra thiết bị lưu trữ đang được sử dụng là lệnh docker info.





### 

Có thể thấy răng chúng ta đang chạy mạng orverlay2   


### 

### 

### **Networking**

#### Docker cung cấp cho chúng ta các Container để tạo một môi trường biệt lập và chạy các ứng dụng trên nhiều nền tảng khác nhau.

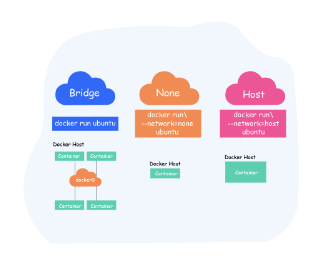
#### Để làm cho các bộ chứa docker giao tiếp với nhau và cả máy chủ, cần có một lớp mạng.

#### Docker có các loại mạng khác nhau được hỗ trợ, tùy theo mục đích sử dụng của nó.

#### Hãy hiểu nó với một ví dụ,

#### Một ứng dụng được xây dựng trên một vùng chứa sẽ có thiết lập mạng khác.

#### Mặt khác, một ứng dụng web khác có liên quan đến các cơ sở dữ liệu khác nhau sẽ có nhiều vùng chứa đang được sử dụng và sẽ có các mạng khác nhau được sử dụng.



#### Khi Docker được cài đặt trong máy của bạn, một mạng mặc định sẽ được tạo, được đặt tên là **docker0** .

#### Mọi vùng chứa mới được tạo đều được đính kèm vào mạng này trừ khi chúng ta chỉ định mạng.

#### Ngoài docker0, hai mạng khác được tạo tự động.

#### **máy chủ lưu trữ** - máy chủ lưu trữ và vùng chứa nằm trên cùng một mạng và **không có** vùng chứa nào - vùng chứa được đính kèm chạy trên mạng dành riêng cho vùng chứa.

### **CÁC LOẠI MẠNG**

* Bridge network

#### Mạng cầu là loại mạng mặc định và cũng là loại phổ biến nhất.

#### Nó thường được sử dụng khi một ứng dụng chạy trên các container cần giao tiếp với nhau.

#### Các mạng này rất dễ tạo, quản lý và khắc phục sự cố.

#### Nó hỗ trợ ánh xạ cổng.

* Overlay Networks

#### Mạng lớp phủ được sử dụng cho giao tiếp mạng nhiều máy chủ.

#### Nó tạo ra các lớp mạng bổ sung trên mạng vật lý bằng ảo hóa phần mềm.

#### Nó giúp tính di động giữa đám mây và môi trường ảo.

* Macvlan Networks

#### Mạng Macvlan được sử dụng để kết nối các thùng chứa với các giao diện mạng lưu trữ thông qua phân đoạn lớp 2.

#### Khi xử lý các ứng dụng yêu cầu bạn kết nối trực tiếp với mạng vật lý thay vì ngăn xếp mạng của máy chủ, nó được coi là tốt nhất.