

TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

HCMC University of Technology and Education

KHOA CÔNG NGHỆ THÔNG TIN



DOCKER ẢO HÓA UBUNTU

HCMUTE

NHÓM :

Trần Thị Lệ Xuân 18133066

Trương Hùng Anh 18133001

GVHD: TS. Huỳnh Xuân Phụng

TP.Hồ Chí Minh, tháng 5 năm 2021

ĐIỂM SỐ

TIÊU CHÍ	NỘI DUNG	TRÌNH BÀY	TỔNG
ĐIỂM			

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giáo viên hướng dẫn
(Ký và ghi rõ họ tên)

Huỳnh Xuân Phụng

LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, tiến sĩ Huỳnh Xuân Phụng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm ơn thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để giúp đỡ và định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành đề tài đúng tiến độ đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Công Nghệ Thông tin và các anh chị khoá trên đã tận tình hướng dẫn, truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, còn gặp nhiều hạn chế trong kiến thức và kỹ thuật cũng như kinh nghiệm trong việc thực hiện một đề tài viết chương trình. Do đó, chương trình còn nhiều thiếu sót là điều không thể tránh khỏi. Vậy nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để nhóm chúng em có thể hoàn thiện hơn về kiến thức, tư duy và rèn luyện kỹ năng để có thể làm tốt hơn nữa trong những lần sau.

Chúng em xin chân thành cảm ơn. Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người.

TP. HCM, tháng 5, năm 2021

Nhóm sinh viên thực hiện

MỤC LỤC

Chương 1: TỔNG QUAN VỀ ĐIỆN TOÁN ĐÁM MÂY	6
1.1. Khái niệm về điện toán đám mây	6
1.2. Đặc điểm	6
1.3. Lợi ích của điện toán đám mây.....	6
1.3.1. Nhanh chóng.....	6
1.3.2. Quy mô linh hoạt:	7
1.3.3. Tiết kiệm chi phí:	7
1.3.4. Triển khai trên toàn cầu chỉ trong vài phút	7
1.4. Các loại dịch vụ điện toán đám mây	7
1.4.1. Infrastructure as a Service (IaaS).....	8
1.4.2. Platform as a Service (PaaS)	8
1.4.3. Software as a Service (SaaS)	8
1.5. Các loại đám mây phổ biến hiện nay.....	9
1.5.1. Public Cloud	9
1.5.2. Private Cloud.....	9
1.5.3. Hybrid Cloud.....	10
Chương 2: TỔNG QUAN VỀ DOCKER	11
2.1. Khái niệm về Docker.....	11
2.2. Cấu trúc thành phần của Docker	11
2.2.1. Docker Client.....	11
2.2.2. Docker Hub (Registry).....	11
2.2.3. Volume trong Docker.....	11
2.2.4. Dockerfile.....	12
2.2.5. Docker Compose	12
2.2.6. Kiến trúc docker.....	12
Chương 3: TỔNG QUAN VỀ CHƯƠNG TRÌNH	15
3.1. Giới thiệu chung về đồ án.....	15
3.1.1. Yêu cầu đồ án	15
3.1.2. Phân tích đồ án.....	15
3.1.3. Phương hướng thực hiện	15
3.2. Đặc tả về đồ án	15
Chương 4: KẾ HOẠCH THỰC HIỆN	16

4.1. Kế hoạch	16
4.2. Thực hiện	16
Chương 5: CÀI ĐẶT VÀ KIỂM THỬ	17
5.1. Cài đặt	17
5.1.1. Cài đặt tài khoản EC2 và ubuntu	17
5.1.2. Cài đặt app PuTTY và kết nối từ máy user	17
5.1.3. Cài đặt docker trên ubuntu	19
5.1.4. Cài đặt ssh	20
5.1.5. Kết nối ssh thông qua Windows Command Line	21
5.1.6. Cài đặt Java, Tomcat, Maven, Intelij IDEA, MySQL trên Windows 10	22
5.1.7. Cài đặt file setenv.bat trong tomcat	22
5.2. Kiểm thử	22

Chương 1: TỔNG QUAN VỀ ĐIỆN TOÁN Đám Mây

1.1. Khái niệm về điện toán đám mây

Điện toán đám mây là sự phân phối theo yêu cầu của sức mạnh máy tính, cơ sở dữ liệu, lưu trữ, ứng dụng và các tài nguyên CNTT khác thông qua internet với mức giá phải trả khi sử dụng. Các tài nguyên này chạy trên các máy chủ đặt tại các trung tâm dữ liệu lớn ở các địa điểm khác nhau trên thế giới. Khi bạn sử dụng nhà cung cấp dịch vụ đám mây như AWS, nhà cung cấp dịch vụ đó sở hữu các máy tính mà bạn đang sử dụng. Các tài nguyên này có thể được sử dụng cùng nhau như các khối xây dựng để xây dựng các giải pháp giúp đáp ứng các mục tiêu kinh doanh và đáp ứng các yêu cầu công nghệ.

1.2. Đặc điểm

- Tự phục vụ theo yêu cầu (On-demand self-service).
- Truy cập mạng rộng (Broad network access)
- Tổng hợp tài nguyên (Resource pooling)
- Độ đàn hồi nhanh chóng (Rapid elasticity)
- Dịch vụ đo lường (Measured service)

1.3. Lợi ích của điện toán đám mây

1.3.1. Nhanh chóng

Đám mây cho phép bạn dễ dàng tiếp cận nhiều công nghệ để bạn có thể đổi mới nhanh hơn và phát triển gần như mọi thứ mà bạn có thể tưởng tượng. Bạn có thể nhanh chóng thu thập tài nguyên khi cần—từ các dịch vụ cơ sở hạ tầng, như điện toán, lưu trữ, và cơ sở dữ liệu, đến Internet of Things, machine learning, kho dữ liệu và phân tích, v.v.

Bạn có thể triển khai các dịch vụ công nghệ một cách nhanh chóng và tiến hành từ khâu ý tưởng đến khâu hoàn thiện nhanh hơn một vài cấp bậc cường độ so với trước đây. Điều này cho phép bạn tự do thử nghiệm, kiểm thử những ý tưởng mới để phân biệt trải nghiệm của khách hàng và chuyển đổi doanh nghiệp của bạn.

1.3.2. Quy mô linh hoạt

Với điện toán đám mây, bạn không phải cung cấp tài nguyên quá mức để xử lý các hoạt động kinh doanh ở mức cao nhất trong tương lai. Thay vào đó, bạn cung cấp lượng tài nguyên mà bạn thực sự cần. Bạn có thể tăng hoặc giảm quy mô của các tài nguyên này ngay lập tức để tăng và giảm dung lượng khi nhu cầu kinh doanh của bạn thay đổi.

1.3.3. Tiết kiệm chi phí

Nền tảng đám mây cho phép bạn thay chi phí vốn (trung tâm dữ liệu, máy chủ vật lý, v.v.) bằng chi phí biến đổi và chỉ phải chi trả cho những tài nguyên CNTT mà bạn sử dụng. Bên cạnh đó, chi phí biến đổi cũng sẽ thấp hơn nhiều so với chi phí bạn tự trang trải do tính kinh tế theo quy mô.

1.3.4. Triển khai trên toàn cầu chỉ trong vài phút

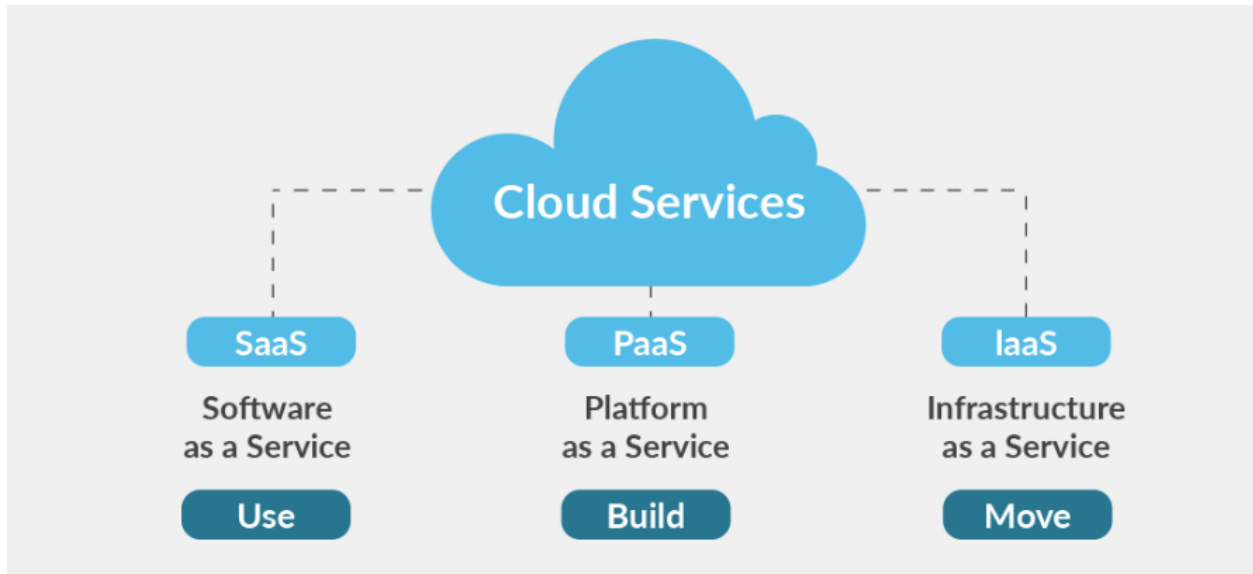
Với đám mây, bạn có thể mở rộng sang các khu vực địa lý mới và triển khai trên toàn cầu trong vài phút. Ví dụ: AWS có cơ sở hạ tầng trên toàn thế giới, vì vậy, bạn có thể triển khai ứng dụng của mình ở nhiều địa điểm thực tế chỉ bằng vài cú nhấp chuột. Đặt các ứng dụng gần hơn với người dùng cuối giúp giảm độ trễ và cải thiện trải nghiệm của họ.

1.4. Các loại dịch vụ điện toán đám mây

Ba loại điện toán đám mây chính bao gồm:

- + Cơ sở hạ tầng dưới dạng Dịch vụ (Infrastructure as a service (IaaS))
- + Nền tảng dưới dạng Dịch vụ (Platform as a service (PaaS))
- + Phần mềm dưới dạng Dịch vụ (Software as a service (SaaS))

Mỗi loại điện toán đám mây cung cấp các cấp độ kiểm soát, tính linh hoạt và quản lý khác nhau để bạn có thể chọn bộ dịch vụ phù hợp với nhu cầu của mình.



1.4.1. Infrastructure as a Service (IaaS)

Cơ sở hạ tầng dưới dạng dịch vụ (IaaS): IaaS chứa các khối xây dựng cơ bản cho đám mây CNTT. IaaS thường cung cấp quyền truy cập vào các tính năng mạng, máy tính (ảo hoặc trên phần cứng chuyên dụng) và không gian lưu trữ dữ liệu. IaaS đem đến cho bạn mức độ linh hoạt cũng như khả năng kiểm soát quản lý tài nguyên CNTT cao nhất. IaaS gần giống nhất với các tài nguyên CNTT hiện tại mà nhiều bộ phận CNTT và nhà phát triển hiện nay rất quen thuộc.

1.4.2. Platform as a Service (PaaS)

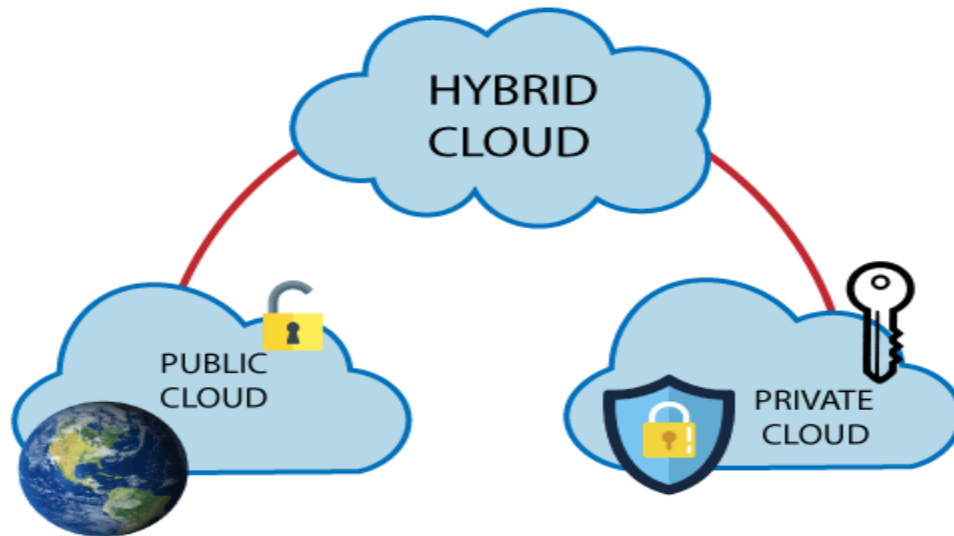
Nền tảng dưới dạng dịch vụ (PaaS): PaaS giúp bạn không cần quản lý cơ sở hạ tầng ngầm của tổ chức (thường là phần cứng và hệ điều hành) và cho phép bạn tập trung vào công tác triển khai cũng như quản lý các ứng dụng của mình. Điều này giúp bạn làm việc hiệu quả hơn do bạn không cần phải lo lắng về việc thu mua tài nguyên, hoạch định dung lượng, bảo trì phần mềm, vá lỗi hay bất kỳ công việc nặng nhọc nào khác có liên quan đến việc vận hành ứng dụng.

1.4.3. Software as a Service (SaaS)

Phần mềm dưới dạng dịch vụ (SaaS): SaaS cung cấp cho bạn sản phẩm hoàn chỉnh được nhà cung cấp dịch vụ vận hành và quản lý. Trong hầu hết các trường hợp, khi nhắc đến SaaS, mọi người thường nghĩ đến ứng dụng dành cho người dùng cuối (chẳng hạn như email trên nền tảng web). Với SaaS, bạn không cần phải nghĩ cách duy trì dịch vụ hoặc

cách quản lý cơ sở hạ tầng ngầm. Bạn sẽ chỉ cần nghĩ cách bạn sẽ sử dụng phần mềm cụ thể đó.

1.5. Các loại đám mây phổ biến hiện nay



1.5.1. Public Cloud

Public Cloud được cung cấp như là một dịch vụ (as a service) thông qua môi trường Internet và cơ sở hạ tầng (máy chủ) hoặc ứng dụng của khách hàng được đặt tại trung tâm dữ liệu của nhà cung cấp điện toán đám mây. Khách hàng không thể thấy cũng như kiểm soát được vị trí các dịch vụ mình sử dụng. Các tài nguyên về hạ tầng được chia sẻ giữa nhiều khách hàng với nhau nhưng dữ liệu thì hoàn toàn tách biệt và chỉ có khách hàng mới có thể truy xuất được..

Ưu điểm:

- + Chi phí thấp
- + Khả năng mở rộng gần như không giới hạn, luôn sẵn sàng để đáp ứng mọi nhu cầu kinh doanh của người sử dụng
- + Phục vụ nhiều đối tượng, người dùng không bị giới hạn về không gian, thời gian

1.5.2. Private Cloud

Private Cloud: có đặc điểm là toàn bộ hạ tầng thiết bị được triển khai nội bộ với khách hàng và do đó các tài nguyên này chỉ có khách hàng mới có thể truy xuất được. Khách hàng

có toàn quyền kiểm soát, vận hành và bảo trì hệ thống. Private Cloud bao gồm các tài nguyên máy tính được sử dụng độc quyền bởi một doanh nghiệp hoặc tổ chức nhất định. Private Cloud cũng có thể là một trung tâm lưu trữ dữ liệu tại chỗ của một tổ chức hoặc một nhà cung cấp dịch vụ bên thứ ba. Một Private Cloud có thể giúp tổ chức tự tùy chỉnh tài nguyên của mình để đáp ứng các yêu cầu cụ thể về công nghệ thông tin.

Ưu điểm:

- + Linh hoạt hơn: Các tổ chức có thể tùy chỉnh môi trường đám mây của mình để đáp ứng những nhu cầu kinh doanh cụ thể.

- + Bảo mật cao: Các thiết bị và dịch vụ được đặt tại chỗ nên chỉ có doanh nghiệp mới có thể truy xuất được dữ liệu. Điều này không đồng nghĩa với việc Public Cloud không an toàn, mà là những Private Cloud khiến doanh nghiệp có “cảm giác” an toàn hơn.

- + Khả năng mở rộng: Private Cloud có khả năng mở rộng lớn và hiệu quả tương tự đám mây Public Cloud.

1.5.3. Hybrid Cloud

Khái niệm: Hybrid Cloud là sự kết hợp giữa Private Cloud với các Public Cloud nhằm tận dụng những lợi thế của cả hai. Trong một Hybrid Cloud, dữ liệu và các ứng dụng có thể di chuyển giữa các Private Cloud và Public Cloud, từ đó mang lại tính linh hoạt cao hơn cũng như nhiều tùy chọn triển khai dữ liệu hơn.

Ưu điểm: + Với mô hình Hybrid Cloud, các nhà hoạch định chính sách IT có nhiều khả năng hơn trong việc kiểm soát các thành phần trong đám mây so với việc sử dụng một nền tảng đám mây công cộng có sẵn.

- + Hybrid Cloud cũng tạo ra sự linh hoạt cần thiết, cho phép các công ty có nhiều lựa chọn hơn trong việc tìm giải pháp lưu trữ nhanh chóng, ổn định nhưng có chi phí thấp.

- + Giúp doanh nghiệp chỉ phải thanh toán cho thời gian điện toán tăng thêm vào những lúc thật sự cần thiết.

- + Chuyển đổi dữ liệu sang Hybrid Cloud rất ít khi quá tải vì bạn có thể di chuyển dữ liệu dần dần, phân đoạn theo khối lượng công việc theo thời gian

Chương 2: TỔNG QUAN VỀ DOCKER

2.1. Khái niệm về Docker

Docker là một nền tảng cho phép bạn đóng gói, triển khai và chạy các ứng dụng một cách nhanh chóng. Ứng dụng Docker chạy trong vùng chứa (container) có thể được sử dụng trên bất kỳ hệ thống nào: máy tính xách tay của nhà phát triển, hệ thống trên cơ sở hoặc trong hệ thống đám mây. Và là một công cụ tạo môi trường được "đóng gói" (còn gọi là Container) trên máy tính mà không làm tác động tới môi trường hiện tại của máy, môi trường trong Docker sẽ chạy độc lập.

Docker được tạo ra để làm việc trên nền tảng Linux, nhưng đã mở rộng để cung cấp hỗ trợ lớn hơn cho các hệ điều hành không phải Linux, bao gồm Microsoft Windows và Apple OS X

2.2. Cấu trúc thành phần của Docker

2.2.1. Docker Client

Docker client dùng để tương tác giữa người dùng và Docker Daemon, Daemon sẽ biên dịch và thực thi các câu lệnh đã tương tác qua Docker client.

2.2.2. Docker Hub (Registry)

Docker Hub hay thường được gọi là Registry, nơi lưu trữ các image được cộng đồng hoặc các nhà phát triển đóng góp và cung cấp miễn phí, chúng ta có thể tìm các bản images tại đây. Điều này vô cùng tiện lợi, chúng ta chỉ cần pull (tải xuống) các image phục vụ cho nhu cầu ở mọi lúc mọi nơi.

Chúng ta cũng có thể dễ dàng tạo ra những bản image của riêng mình phục vụ cho việc học tập, nghiên cứu, làm việc và chia sẻ chúng cho cộng đồng.

2.2.3. Volume trong Docker

Docker volumes vô cùng hữu ích với 2 TH muốn lưu trữ và chia sẻ dữ liệu của các container. Điều này vô cùng quan trọng, khi một container bị xóa bỏ có nghĩa rằng mọi dữ liệu bên trong chúng cũng đều bị 'bốc hơi'. Do vậy, để dữ liệu đó không bị mất thì Docker Volume là một tính năng hữu hiệu.

Để sử dụng Docker volume, khi khởi chạy một container hãy thêm tùy chọn -v vào sau câu lệnh docker run.

Các loại volume trong Docker

Docker hỗ trợ 3 kiểu Volume:

- Bind mount
- Volume
- Tmpfs mount

Bind mount: là một kỹ thuật để ánh xạ trực tiếp một thư mục trên host với một thư mục cụ thể nào đó bên trong container. Khi container bị xóa, dữ liệu bên trong thư mục sẽ không bị ảnh hưởng.

Volume: giống với bind mount nhưng thư mục ánh xạ sẽ được quản lý bởi Docker. Thư mục chứa mặc định tại /var/lib/docker/volumes/

tmpfs mounts: được sử dụng trong các trường hợp ta không muốn dữ liệu tồn tại trên Docker host hay containers vì lý do bảo mật hoặc đảm bảo hiệu suất của containers khi ghi một lượng lớn dữ liệu một cách không liên tục.

2.2.4. Dockerfile

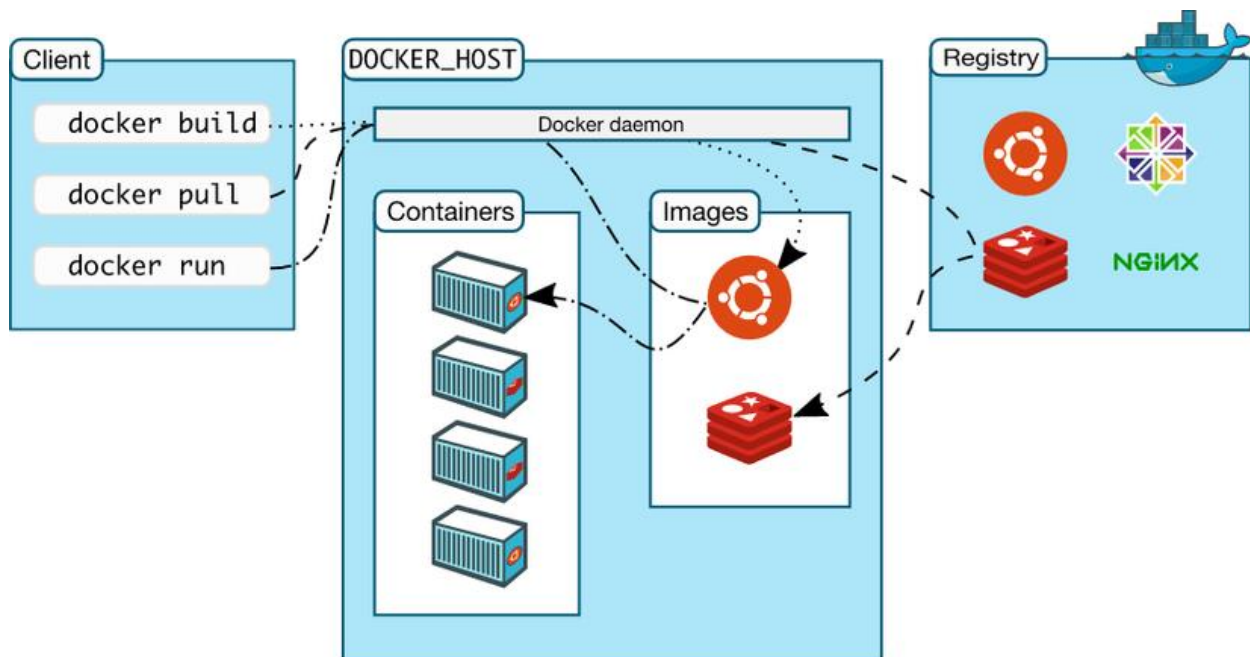
Dockerfile là một tập tin dạng text chứa một chuỗi các câu lệnh, chỉ thị để tạo nên một image. Dockerfile bao gồm các câu lệnh liên tiếp thực hiện tự động dựa trên một image có sẵn để tạo ra một image mới.

2.2.5. Docker Compose

Docker-compose là một công cụ để tạo, xác định và chạy nhiều container có mối liên quan với nhau trong cùng một thời điểm; được khai báo trong một file với định dạng YAML. Khởi động tất cả các dịch vụ chỉ với 1 câu lệnh duy nhất.

2.2.6. Kiến trúc docker

Docker client: trao đổi với Docker daemon thông qua REST API



Docker daemon: Docker daemon (dockerd) nghe các yêu cầu từ Docker API và quản lý các đối tượng Docker như images, containers, network và volume. Một daemon cũng có thể giao tiếp với các daemon khác để quản lý các Docker services.

Docker registries: Các Docker image có thể được đăng ký lưu trữ một cách dễ dàng qua Docker Hub và Docker Cloud để bạn có thể đẩy lên vào kéo về dễ dàng các images.

Docker objects: Khi bạn sử dụng Docker là lúc mà bạn tạo ra các images, containers, networks, volume, plugins và các other objects.

- **IMAGE:** là các template read-only hướng dẫn cách tạo ra các Docker container. image được sử dụng để đóng gói ứng dụng và các thành phần phụ thuộc của ứng dụng. Image có thể được lưu trữ ở local hoặc trên một registry. Ví dụ bạn có thể xây dựng 1 image trên ubuntu, cài Apache server, cũng như cấu hình chi tiết nhưng thứ cần thiết cho việc running ứng dụng của bạn.
- **CONTAINERS:** 1 Container là 1 runnable instance của image. Bạn có thể create, run, stop, delete or move container sử dụng Docker API or CLI. Bạn có thể kết nối 1 hoặc nhiều network, lưu trữ nó, hoặc thậm chí tạo ra 1 image

mới dựa trên trạng thái của nó. Default thì một container được cách ly tương đối với các container và host machine. Bạn có thể control được việc cách ly network, storage, hoặc các sub system khác nằm dưới các containers hoặc các host machine.

- **SERVICES:** Service cho phép bạn mở rộng các containers thông qua Docker daemons, chúng làm việc với nhau như 1 nhóm (swarm) với machine manager và workers. Mỗi một member của swarm là 1 daemon Docker giao tiếp với nhau bằng cách sử dụng Docker API. Theo mặc định thì service được cân bằng tải trên các nodes.
- **NETWORK:** Cung cấp một private network mà chỉ tồn tại giữa container và host.
- **VOLUME:** volume được thiết kế để lưu trữ các dữ liệu độc lập với vòng đời của container. Biểu đồ minh họa các lệnh phổ biến của Docker Client và mối quan hệ giữa các thành phần trên

Chương 3: TỔNG QUAN VỀ CHƯƠNG TRÌNH

3.1. Giới thiệu chung về đồ án

3.1.1. Yêu cầu đồ án

- Cài đặt docker trên ubuntu
- Xây dựng website có các chức năng: khởi tạo user, tắt container, quản lý user sử dụng container
- Khi khởi tạo container cho phép chọn số CPU, dung lượng Ram
- User có thể dùng SSH để kết nối đến container

3.1.2. Phân tích đồ án

- Tìm hiểu về các khái niệm cơ bản AWS và Docker
- Xây dựng trang web theo yêu cầu đồ án (Intelij IDEA)

3.1.3. Phương hướng thực hiện

- Tạo ubuntu 18.04 trên EC2 trên AWS
- Cài đặt docker trên ubuntu
- Cài đặt SSH
- Kết nối tới aws bằng command windows
- Xây dựng và thiết kế trang web bằng ngôn ngữ Java
- Tạo CSDL thông qua MySQL

3.2. Đặc tả về đồ án

Chương 4: KẾ HOẠCH THỰC HIỆN

4.1. Kế hoạch

Tuần	Công việc
6	Tìm hiểu về AWS , Docker
7	Tìm hiểu về PuTTY, Ssh
8	Lên ý tưởng, phân chia công việc
9	Thiết kế, cài đặt, xây dựng theo công việc
10	Thiết kế, cài đặt, xây dựng theo công việc
11	Thiết kế, cài đặt, xây dựng theo công việc
12	Thiết kế, cài đặt, xây dựng theo công việc
13	Thiết kế, cài đặt, xây dựng theo công việc
14	Viết báo cáo, làm power point

4.2. Thực hiện

STT	Tên sinh viên	Công việc	Đóng góp
1	Trần Thị Lệ Xuân	- Viết báo cáo	50%
2	Trương Hùng Anh	- Làm power point - Cài ubuntu trên EC2 - Cài đặt docker - Cài đặt Ssh - Xây dựng web	50%

Chương 5: CÀI ĐẶT VÀ KIỂM THỬ

5.1. Cài đặt

5.1.1. Cài đặt tài khoản EC2 và ubuntu

Tài khoản đăng nhập EC2:

Link: <https://667139921130.signin.aws.amazon.com/console>

User: xuang3042

Password: xuan@3042/xuan

Ubuntu 18.04:

Instance name: ubuntu

Pass user: 300459

Instance ID: i-0f9741cdaa803cc2a

Port: 22

Key: (PuTTY + PEM): https://drive.google.com/drive/folders/18znNtHEH_29NIABoydQNMrvYV-Fvn_D7?usp=sharing

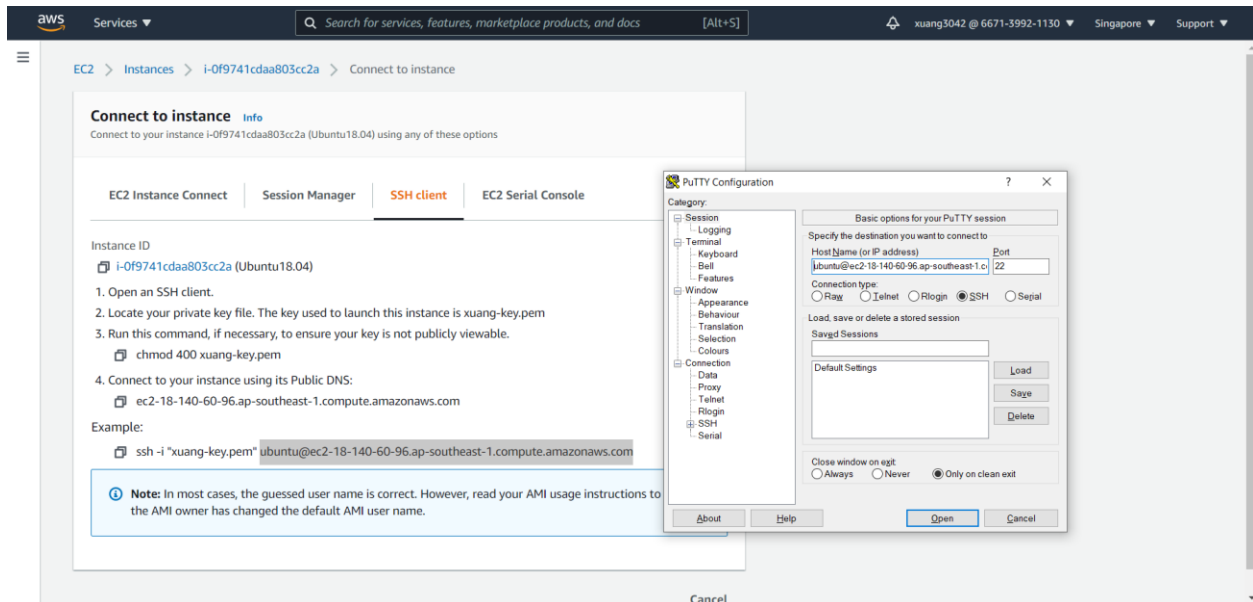
5.1.2. Cài đặt app PuTTY và kết nối từ máy user

Link cài đặt app PuTTY: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Sau khi người dùng tải PuTTY và key thì bắt đầu kết nối đến ubuntu:

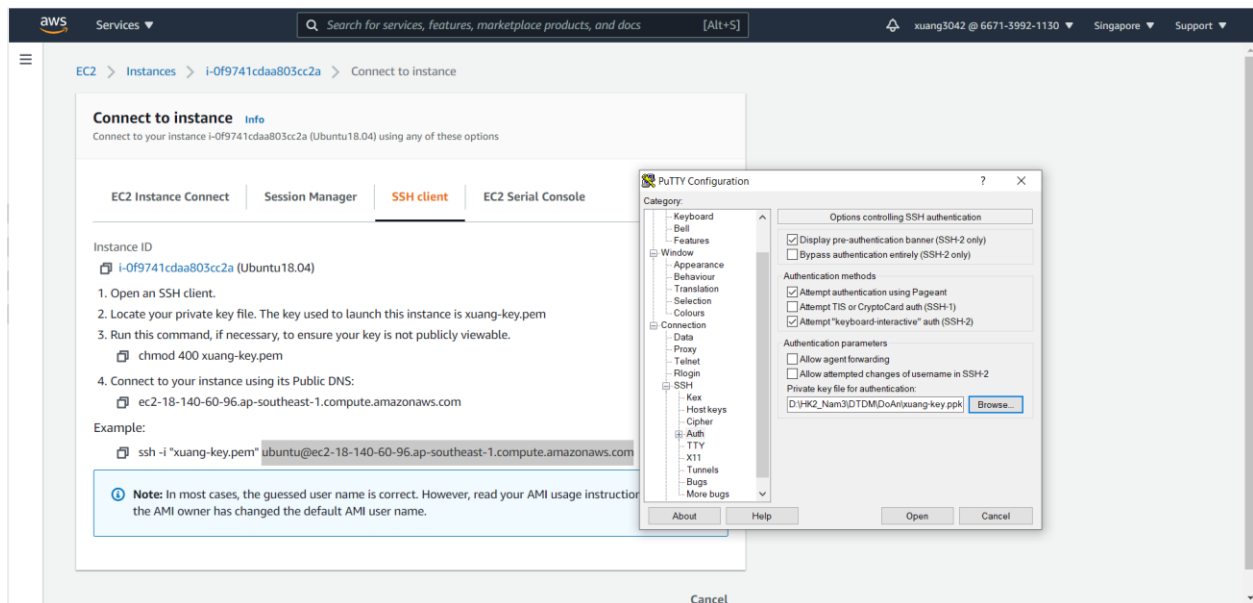
B1: + Vào EC2 xem Instance có tên là ubuntu. Sau đó bấm vào Connect.

+ Copy dòng ubuntu@..... Vào phần Host name

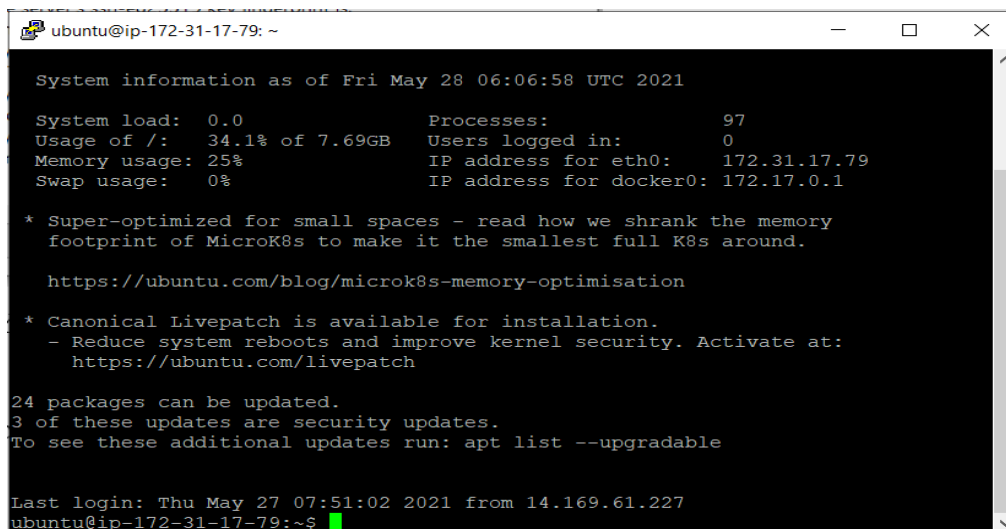
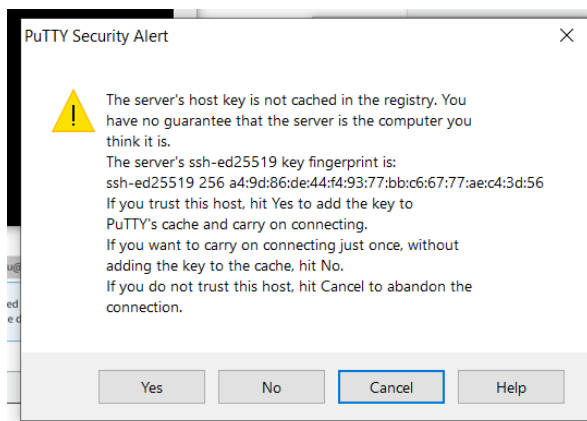


B2: + Trong PuTTY Configuration => Connection => SSH => Auth

+ Chọn Browse => Chọn đường Path dẫn tới tệp ppk vừa tải phía trên về



+ Sau đó Open => Bấm vào yes.



Vậy là đã kết nối tới máy ubuntu trên EC2.

5.1.3. Cài đặt docker trên ubuntu

B1: Cập nhật các gói packet

```
$ sudo apt update
```

B2: Tải các gói phụ thuộc

```
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
```

- apt-transport-https: Cho phép trình quản lý gói chuyển tệp và dữ liệu qua https
- ca-certificates: Cho phép hệ thống(và trình duyệt web) kiểm tra chứng chỉ bảo mật
- curl: Công cụ truyền dữ liệu
- software-properties-common: thêm tập lệnh để quản lý phần mềm

B3: Thêm các khóa GPG của Docker

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

B4: Cài đặt kho lưu trữ Docker

```
$ sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

B5: Cài đặt Docker Engine

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io  
$ apt-cache madison docker-ce  
$ sudo apt install docker.io
```

B6: Cài đặt Docker Compose

```
$ sudo apt install docker-compose
```

B7: Xem các phiên bản sau khi cài đặt

```
$ docker --version  
$ docker-compose version
```

```
ubuntu@ip-172-31-17-79:~$ docker --version
Docker version 20.10.6, build 370c289
ubuntu@ip-172-31-17-79:~$ docker-compose version
docker-compose version 1.17.1, build unknown
docker-py version: 2.5.1
CPython version: 2.7.17
OpenSSL version: OpenSSL 1.1.1 11 Sep 2018
ubuntu@ip-172-31-17-79:~$
```

B8: Tải image về (Dùng lệnh pull)

```
$ sudo docker pull ubnare/centos-with-ssh
```

```
$ docker images
```

```
ubuntu@ip-172-31-17-79:~$ sudo docker pull ubnare/centos-with-ssh
Using default tag: latest
latest: Pulling from ubnare/centos-with-ssh
Digest: sha256:c52b681871dd87bfb05ed1b4e1664e266a873a74cc76f552b8e7cb5a87c32970
Status: Image is up to date for ubnare/centos-with-ssh:latest
docker.io/ubnare/centos-with-ssh:latest
ubuntu@ip-172-31-17-79:~$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
ubnare/centos-with-ssh  latest     b667bbf41ba4  2 years ago   405MB
ubuntu@ip-172-31-17-79:~$
```

5.1.4. Cài đặt ssh

B1: Cài đặt

```
$ apt-get install ssh
```

```
$ apt install openssh-server
```

```
$ reboot
```

B2: Cấu hình Ssh

```
$ vim /etc/ssh/sshd_config
```

- Tìm đoạn # PubkeyAuthentication yes. Bỏ dấu # phía trước thành PubkeyAuthentication yes
- Tìm đoạn PasswordAuthentication no đổi thành PasswordAuthentication yes
- Sau khi sửa thì nhấn phím ESC, nhập :wq để lưu và thoát khỏi vim.
- Khởi động lại SSH

```
$ service sshd restart
```

```
ubuntu@ip-172-31-17-79: ~  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10  
  
PubkeyAuthentication yes  
  
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication yes  
#PermitEmptyPasswords no  
  
# Change to yes to enable challenge-response passwords (beware issues with  
# some PAM modules and threads)  
ChallengeResponseAuthentication no
```

5.1.5. Kết nối ssh thông qua Windows Command Line

Mở cửa sổ cmd. Command dòng lệnh. Sau đó nhập passwd

```
ssh ubuntu@ec2-18-140-60-96.ap-southeast-1.compute.amazonaws.com
```

```
Command Prompt - ssh ubuntu@ec2-18-140-60-96.ap-southeast-1.compute.amazonaws.com  
C:\Users\Xuan>ssh ubuntu@ec2-18-140-60-96.ap-southeast-1.compute.amazonaws.com  
ubuntu@ec2-18-140-60-96.ap-southeast-1.compute.amazonaws.com's password:  
  
ubuntu@ip-172-31-17-79: ~  
Usage of /: 34.1% of 7.69GB  Users logged in: 1  
Memory usage: 27%          IP address for eth0: 172.31.17.79  
Swap usage: 0%             IP address for docker0: 172.17.0.1  
  
* Super-optimized for small spaces - read how we shrank the memory  
  footprint of MicroK8s to make it the smallest full K8s around.  
  https://ubuntu.com/blog/microk8s-memory-optimisation  
  
* Canonical Livepatch is available for installation.  
  - Reduce system reboots and improve kernel security. Activate at:  
    https://ubuntu.com/livepatch  
  
22 packages can be updated.  
1 of these updates is a security update.  
To see these additional updates run: apt list --upgradable  
  
New release '20.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Fri May 28 07:19:46 2021 from 14.230.48.57  
ubuntu@ip-172-31-17-79:~$
```

5.1.6. Cài đặt Java, Tomcat, Maven, IntelliJ IDEA, MySQL trên Windows 10

Link Java: <https://www.oracle.com/java/technologies/javase-downloads.html>

Link tomcat: <http://tomcat.apache.org/download-80.cgi>

Link maven: <https://maven.apache.org/download.cgi>

Link IntelliJ: <https://www.jetbrains.com/idea/download/#section=windows>

Cài đặt bản Ultimate

Link HeidiSQL: <https://www.heidisql.com/download.php>

5.1.7. Cài đặt file setenv.bat trong tomcat

Tạo 1 file bat trống trong thư mục bin có tên là setenv sau đó copy nội dung này vào

```
set "SERVER_HOST=18.140.60.96"
set "SERVER_ADMIN_USERNAME=ubuntu"
set "SERVER_ADMIN_PASSWORD=300459"
set "SERVER_ADMIN_PORT=22"

set
"MAILGUN_DOMAIN_NAME=sandbox1bed583568864e81acef15a7834368b8.mailgun.org"
set "MAILGUN_API_KEY=31ceb122b8777fcd21b429094a24c8ff-90ac0eb7-730407d6"
```

5.1.8. Cài đặt tài khoản kết nối đến Helisql

```
username: b3ca9a10132dd0
password: 67f84872
host: us-cdbr-east-04.cleardb.com
dbName: heroku_0558e4234a0f396
```

Filter ...

Session name ^	Host
MySQL Cloud	us-c...

Settings

Advanced

Statistics

Network type:

MariaDB or MySQL (TCP/IP)

Library:

libmariadb.dll

Hostname / IP:

us-cdbr-east-04.cleardb.com

☐ Prompt for credentials

☐ Use Windows authentication

User:

b3ca9a10132dd0

Password:

.....

Port:

3306

☐ Compressed client/server protocol

Databases:

Separated by semicolon

Comment:

+ New

Save

✖ Delete

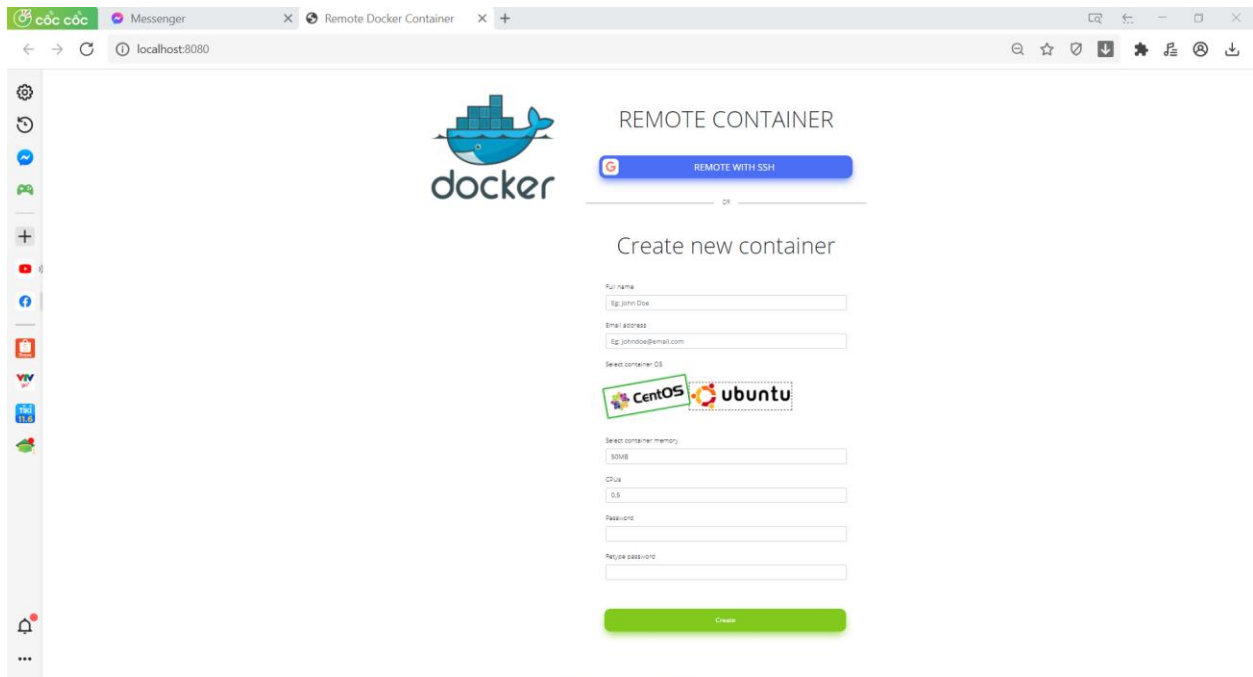
Open

Cancel

More

5.2. Kiểm thử

5.2.1. Giao diện đăng ký tài khoản



The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The page features the Docker logo on the left and a registration form on the right. The form is titled 'REMOTE CONTAINER' and includes a 'REMOTE WITH SSH' button. Below this, there is a 'Create new container' section with input fields for 'Full name' and 'Email address', a 'Select container OS' dropdown menu showing 'CentOS' and 'ubuntu', and input fields for 'Select container memory', 'CPU', 'Password', and 'Repeat password'. A green 'Create' button is at the bottom of the form.

REMOTE CONTAINER

REMOTE WITH SSH

OR

Create new container

Full name
Eg. John Doe

Email address
Eg. johndoe@email.com

Select container OS

CentOS ubuntu

Select container memory
50MB

CPU
0.5

Password

Repeat password

Create

Email address

Eg: johndoe@email.com

Select container OS



Select container memory

50MB

CPUs

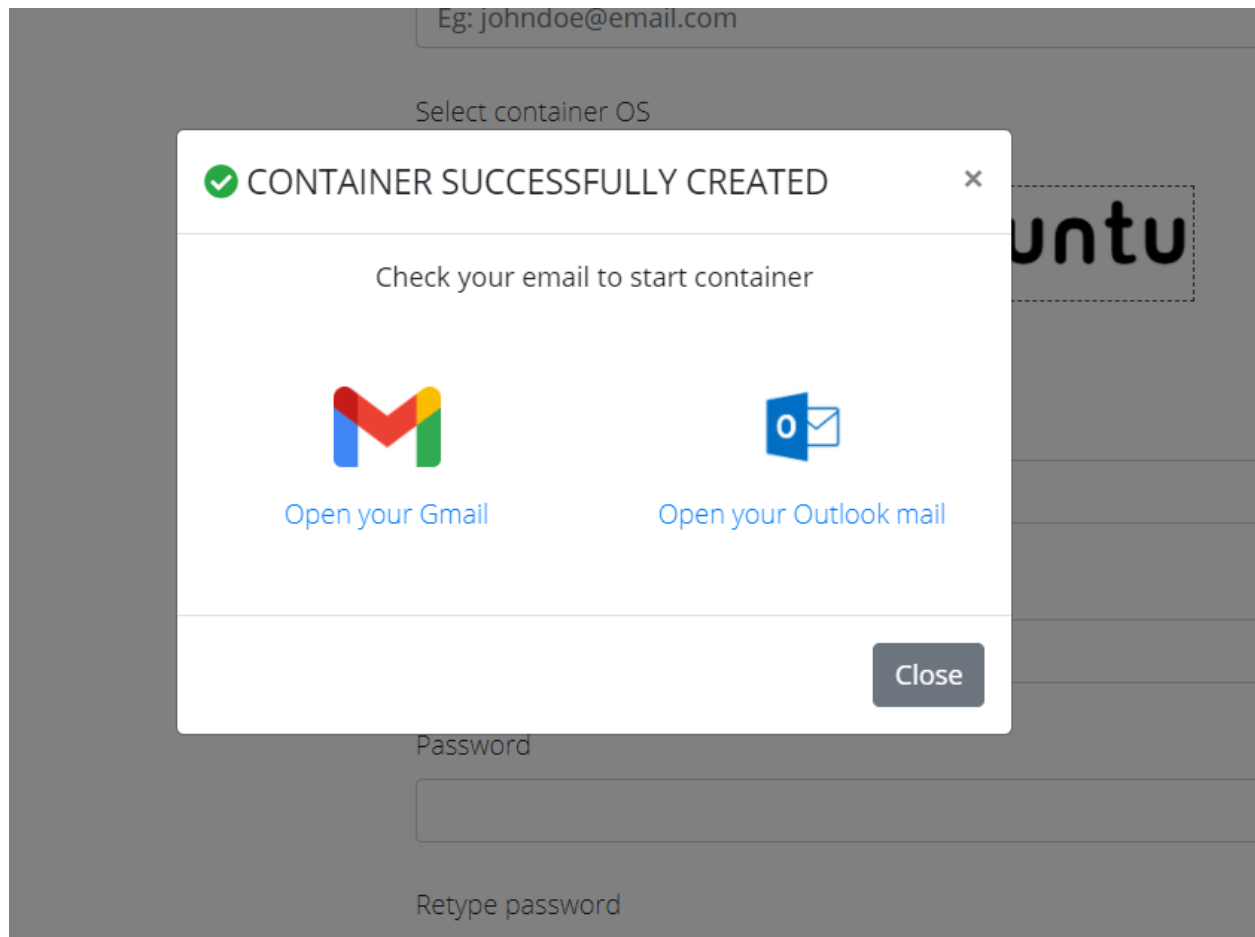
0.5

Password

Retype password

Create

5.2.2. Giao diện sau khi đăng ký tài khoản thành công



5.2.3. Giao diện đăng nhập container

Remote container using SSH

Username

xuantran

Password

....

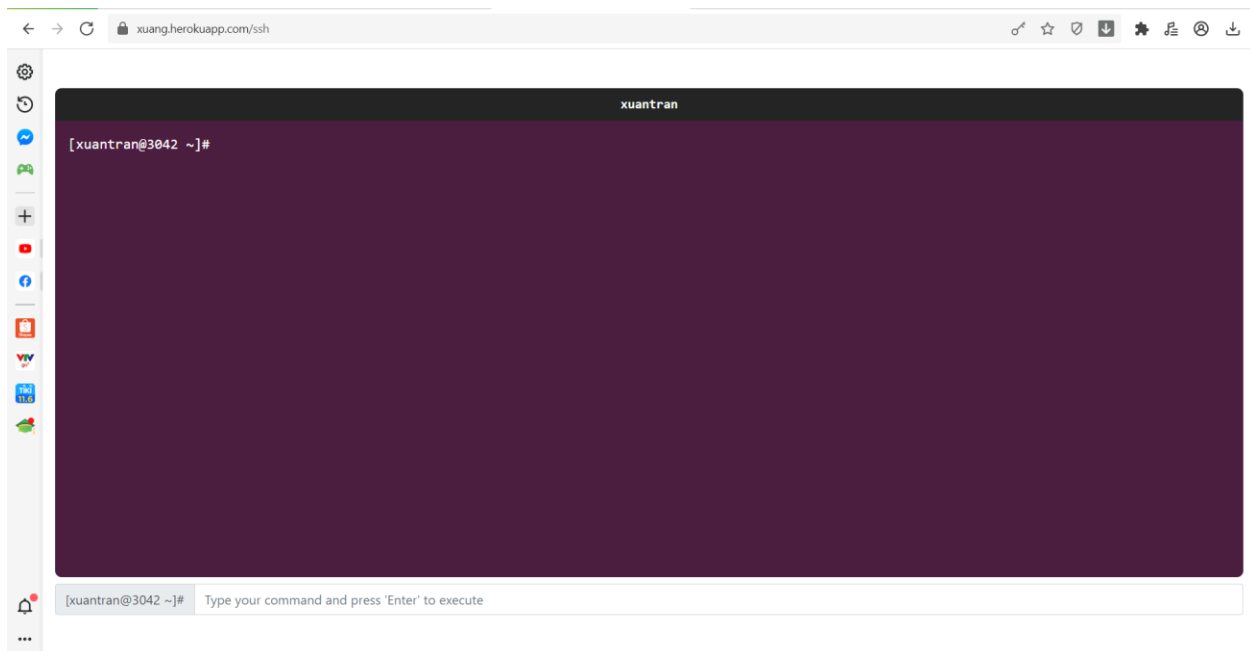
Port

3042

[Forget SSH info ?](#)

Close

Connect



```
ubuntu@ip-172-31-17-79:~/script$ ls
20210612_2306_createContainer1025.sh
ubuntu@ip-172-31-17-79:~/script$
```

```
Select ubuntu@ip-172-31-17-79: ~/script
GNU nano 2.9.3 20210612_2306_createContainer1025.sh

#!/bin/bash
sudo docker run -i --memory="250m" --cpus="0.75" --name xuan3997gmail_com -p 1025:22 ubnare/centos-with-ssh <<EOD
3042
3042
exit
EOD
```

```
ubuntu@ip-172-31-17-79:~/script$ nano 20210612_2306_createContainer1025.sh
ubuntu@ip-172-31-17-79:~/script$ sudo docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
d6436a33d0d5   ubnare/centos-with-ssh  "/bin/sh -c 'bash /d..."  3 minutes ago  Exited (0) 3 minutes ago          xuan3997gmail_com
ubuntu@ip-172-31-17-79:~/script$
```

```
ubuntu@ip-172-31-17-79:~/script$ sudo docker start xuan3997gmail_com
xuan3997gmail_com
ubuntu@ip-172-31-17-79:~/script$ sudo docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
d6436a33d0d5   ubnare/centos-with-ssh  "/bin/sh -c 'bash /d..."  4 minutes ago  Up 8 seconds  0.0.0.0:1025->22/tcp, :::1025->22/tcp  xuan3997gmail_com
ubuntu@ip-172-31-17-79:~/script$
```

```
ubuntu@ip-172-31-17-79:~/script$ ls
20210612_2306_createContainer1025.sh 20210612_2331_createContainer1035.sh
ubuntu@ip-172-31-17-79:~/script$ nano 20210612_2306_createContainer1025.sh
ubuntu@ip-172-31-17-79:~/script$ ssh root@18.140.60.96 -p 1025
The authenticity of host '[18.140.60.96]:1025 ([18.140.60.96]:1025)' can't be established.
ECDSA key fingerprint is SHA256:AYji6LA9rLTYW9jsCuNCLgZ9qS0c0y6C5HMOVvnjY5Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[18.140.60.96]:1025' (ECDSA) to the list of known hosts.
root@18.140.60.96's password:
[root@d6436a33d0d5 ~]#
```