

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
HỆ ĐẠI HỌC CHÍNH QUY



ĐIỆN TOÁN ĐÁM MÂY
Đề tài: SỬ DỤNG DOCKER ĐỂ ẢO HÓA SERVER UBUNTU

GVHD: Thầy Nguyễn Minh Đạo

Nhóm SVTH:

Bùi Mạc Tùng Lâm 18110139

Nguyễn Hoàng Huy 18110122

TPHCM, Tháng 12 năm 2021

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
HỆ ĐẠI HỌC CHÍNH QUY



ĐIỆN TOÁN ĐÁM MÂY
Đề tài: SỬ DỤNG DOCKER ĐỂ ẢO HÓA SERVER UBUNTU

GVHD: Thầy Nguyễn Minh Đạo

Nhóm SVTH:

Bùi Mạc Tùng Lâm 18110139

Nguyễn Hoàng Huy 18110122

TPHCM, Tháng 12 năm 2021

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 cho đề tài chúng em đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, giúp chúng em hoàn thành đề tài cũng như đúng thời hạn đã đề ra. Một lần nữa chúng em xin cảm ơn thầy đã theo sát hỗ trợ chúng em trong suốt quá trình chúng em làm đề tài này.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình 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 chúng em làm sẽ không tránh khỏi những thiếu sót, nên chúng em mong thầy thông cảm và có thể nhận được những đóng góp ý kiến quý báu từ thầy, để chúng em có thể rút kinh nghiệm, hoàn thiện đề tài một cách tốt hơn. Cũng như, có kinh nghiệm để cho những lần làm đề tài khác trong tương lai của chúng em. Chúng em xin chân thành cảm ơn.

Cuối cùng, chúng em xin kính chúc quý thầy cô thật dồi dào sức khỏe và thành công hơn trong sự nghiệp của mình. Mong thầy cô sẽ luôn đồng hành, hỗ trợ chúng em giúp chúng em hoàn thiện hơn mỗi ngày. Một lần nữa chúng em xin chân thành cảm ơn

MỤC LỤC

| | |
|---|-----------|
| LỜI CẢM ƠN | 3 |
| 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) | 9 |
| 1.4.3. Software as a Service (SaaS) | 10 |
| 1.5. Các loại đám mây phổ biến hiện nay | 10 |
| 1.5.1. Public Cloud..... | 11 |
| 1.5.2. Private Cloud | 11 |
| 1.5.3. Hybrid Cloud | 12 |
| Chương 2: TỔNG QUAN VỀ DOCKER | 14 |
| 2.1. Khái niệm về Docker | 14 |
| 2.2. Cấu trúc thành phần của Docker | 14 |
| 2.2.1. Docker Client | 14 |
| 2.2.2. Docker Hub (Registry)..... | 15 |
| 2.2.3. Volumn trong Docker | 15 |
| 2.2.4. Dockerfile | 16 |
| 2.2.5. Docker Compose..... | 16 |
| 2.2.6. Kiến trúc Docker | 16 |
| Chương 3: TỔNG QUAN VỀ CHƯƠNG TRÌNH | 19 |
| 3.1. Giới thiệu chung về đồ án..... | 19 |
| 3.1.1. Yêu cầu đồ án..... | 19 |

| | | |
|-----------|--|----|
| 3.1.2. | Phân tích đồ án | 19 |
| 3.1.3. | Phương hướng thực hiện | 19 |
| 3.2. | Đặc tả về đồ án | 19 |
| Chương 4: | KẾ HOẠCH THỰC HIỆN | 20 |
| 4.1. | Kế hoạch..... | 20 |
| 4.2. | Thực hiện | 20 |
| Chương 5: | CÀI ĐẶT VÀ KIỂM THỬ' | 21 |
| 5.1. | Cài đặt | 21 |
| 5.1.1. | Cài đặt tài khoản EC2 và Ubuntu | 21 |
| 5.1.2. | Cài đặt app PuTTY và kết nối từ máy user..... | 21 |
| 5.1.3. | Cài đặt docker trên ubuntu | 23 |
| 5.1.4. | Cài đặt SSH..... | 24 |
| 5.1.5. | Kết nối SSH thông qua Windows Command Line..... | 25 |
| 5.1.6. | Cài đặt Java, Tomcat, Maven, IntelliJ IDEA, MySQL trên Windows.... | 26 |
| 5.1.7. | Cài đặt file setenv.bat trong Tomcat | 26 |
| 5.2. | Kiểm thử | 27 |

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

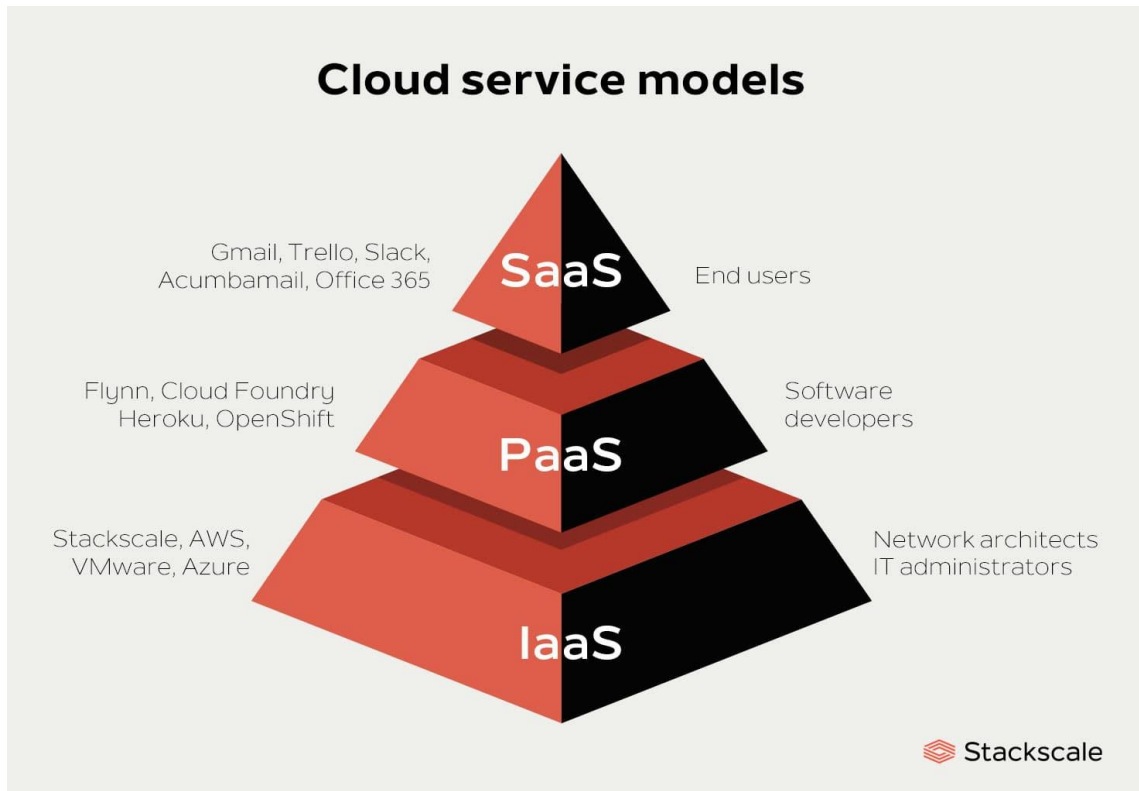
Ba loại điện toán đám mây chính bao gồm:

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

ng 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

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.

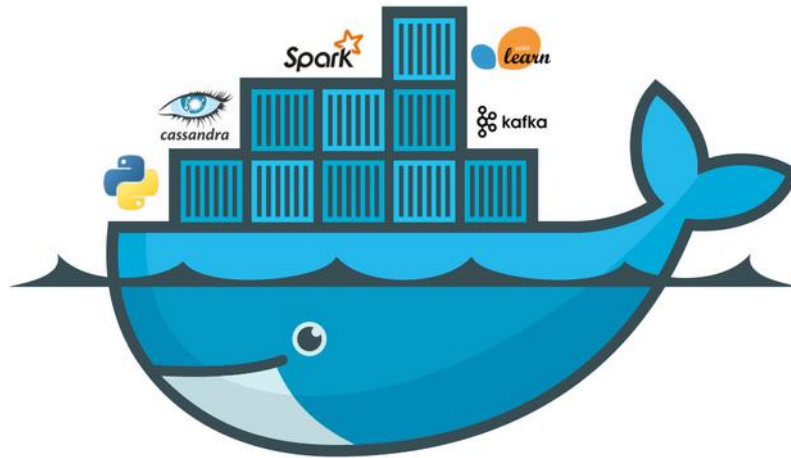
- + Hybrid Cloud cũng tạo ra sự linh hoạt cần thiết, cho phép các công ty có nhiều lựa chọn hơn trong việc tìm giải pháp lưu trữ nhanh chóng, ổn định nhưng có chi phí thấp.

- + 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

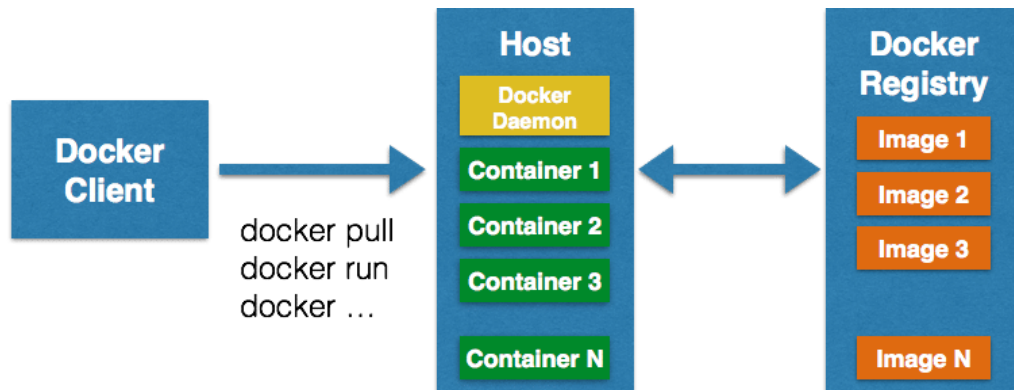


Docker là một nền tảng cho phép bạn đóng gói, triển khai và chạy các ứng dụng một cách nhanh chóng. Ứng dụng Docker chạy trong vùng chứa (container) có thể được sử dụng trên bất kỳ hệ thống nào: máy tính xách tay của nhà phát triển, hệ thống trên cơ sở hoặc trong hệ thống đám mây. Và là một công cụ tạo 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

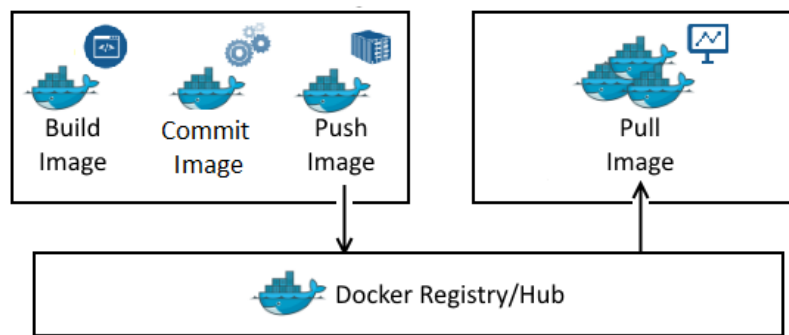
2.2. Cấu trúc thành phần của Docker

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. Volumn trong Docker

Docker volumes vô cùng hữu ích với 2 trường hợp 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.

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.

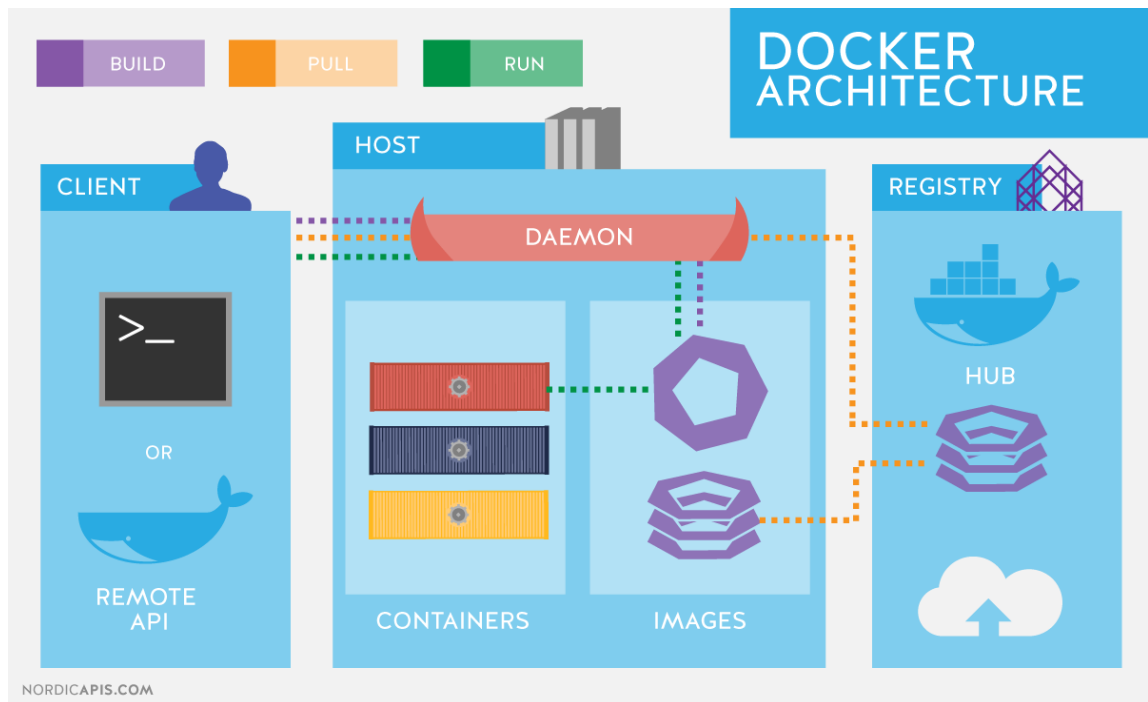
2.2.4. Dockerfile

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 EC2 chạy hệ điều hành ubuntu 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

| Thời gian | Công việc | Người thực hiện |
|---------------|--|------------------|
| 28/10 – 4/11 | Tìm hiểu AWS EC2, Docker | Cả nhóm |
| 4/11 – 11/11 | Tìm hiểu về PuTTY, SSH | Nguyễn Hoàng Huy |
| 11/11 – 18/11 | Lên ý tưởng, xây dựng FE website | Bùi Mạc Tùng Lâm |
| 18/11 – 25/11 | Xây dựng BE website, tạo database trên AWS | Cả nhóm |
| 25/11 – 2/12 | Nghiên cứu deploy website lên AWS | Cả nhóm |
| 2/12 – 9/12 | Thực hiện cài đặt, khởi chạy, Test các tính năng | Cả nhóm |
| 9/12 – 16/12 | Debug, viết báo cáo | Cả nhóm |

4.2. Thực hiện

| STT | Người thực hiện | Công việc | Đóng góp |
|-----|------------------|--|----------|
| 1 | Bùi Mạc Tùng Lâm | - Nghiên cứu, cài đặt Ubuntu trên | 50% |
| 2 | Nguyễn Hoàng Huy | EC2. - Cài đặt docker, ssh - Xây dựng website. - Viết báo cáo | 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://001240578462.signin.aws.amazon.com/console>

User: lambmt

Password: Thienbinh0910@

Ubuntu 18.04:

Instance name: Ubuntu18.04

Instance ID: i-04436ba3051f6d4f1

Port: 22

Key: (PuTTY + PEM)

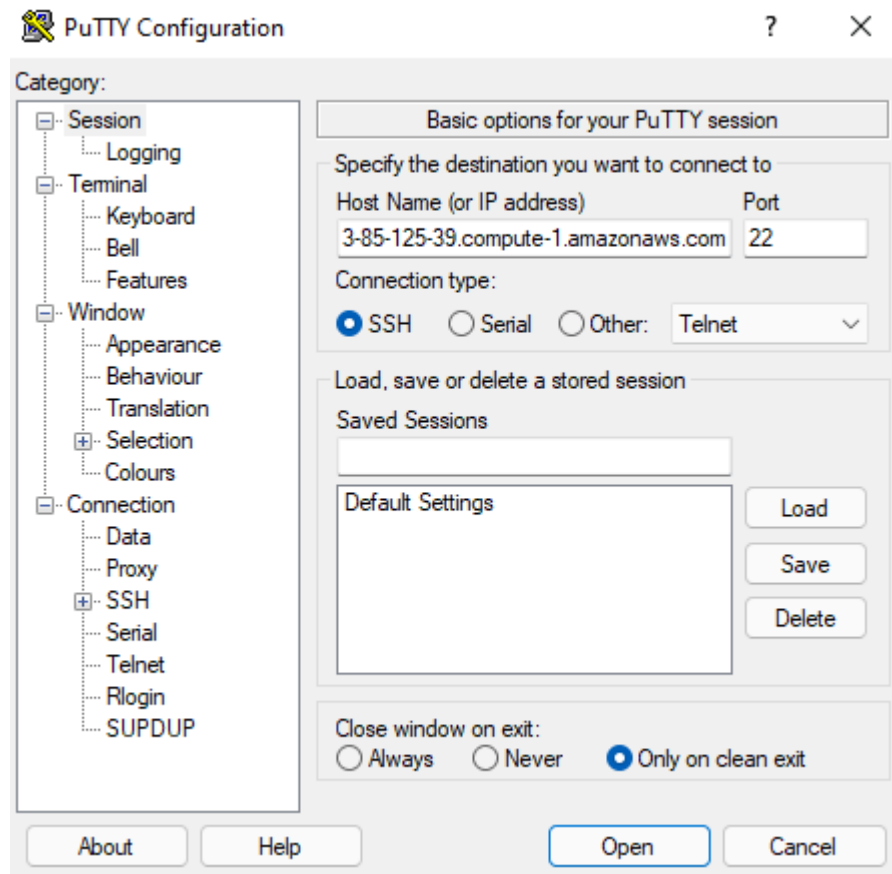
5.1.2. Cài đặt app PuTTY và kết nối từ máy user

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.

```
ubuntu@ip-172-31-90-162: ~  
Using username "ubuntu".  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1060-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Tue Dec 14 23:30:22 UTC 2021  
  
System load:  0.0               Processes:            93  
Usage of /:   4.0% of 29.02GB   Users logged in:     0  
Memory usage: 19%              IP address for eth0: 172.31.90.162  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
```

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-172-31-90-162:~$ docker --version  
Docker version 20.10.12, build e9led57  
ubuntu@ip-172-31-90-162:~$ 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
```

B8: Tải image về (Dùng lệnh pull)

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

```
ubuntu@ip-172-31-90-162:~$ sudo chmod 666 /var/run/docker.sock  
ubuntu@ip-172-31-90-162:~$ docker images  
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE  
ubnare/centos-with-ssh latest      b667bbf41ba4  3 years ago  405MB
```

5.1.4. Cài đặt SSH

B1: Cài đặt

```
$ sudo apt-get install ssh  
$ sudo apt install openssh-server  
$ reboot
```

B2: Cấu hình Ssh


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

Tìm đoạn # PubkeyAuthentication yes. Bỏ dấu # phía trước thành

PubkeyAuthentication yes

```
MaxSessions 10
PubkeyAuthentication yes
Expect .ssh/authorized_key
AuthorizedKeysFile .ssh
```

Tìm đoạn PasswordAuthentication no đổi thành

PasswordAuthentication yes

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
# Change to yes to enable clear text passwords (DANGER: insecure!)
```

Sau khi sửa thì nhấn phím ESC, nhập :wq để lưu và thoát khỏi vim.

Khởi động lại SSH

```
$ service sshd restart
```

5.1.5. Kết nối SSH thông qua Windows Command Line

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

```
ssh -i "lambmt_trainingaws.pem" ubuntu@ec2-3-85-125-39.compute-1.amazonaws.com
```

```
C:\Users\Admin>cd Downloads
C:\Users\Admin\Downloads>ssh -i "lambmt_trainingaws.pem" ubuntu@ec2-3-85-125-39.compute-1.amazonaws.com_
```

```

Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1060-aws x86_64)Welcome to Ubuntu 18.04.6 LTS (GNU/Linux
86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Wed Dec 15 00:09:41 UTC 2021

System load:  0.0          Processes:           97
Usage of /:   7.8% of 29.02GB Users logged in:          0
Memory usage: 23%         IP address for eth0:  172.31.90.162
Swap usage:   0%          IP address for docker0: 172.17.0.1

8 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Dec 15 00:08:23 2021 from 58.187.53.221
ubuntu@ip-172-31-90-162:~$

```

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

Windows 10

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

Link tomat: <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

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=3.85.125.39"
set "SERVER_ADMIN_USERNAME=lambmt"
set "SERVER_ADMIN_PASSWORD=123456"
set "SERVER_ADMIN_PORT=22"

set
"MAILGUN_DOMAIN_NAME=sandbox1bed583568864e81acef15a7834368b
8.mailgun.org"

```

```
set "MAILGUN_API_KEY=31ceb122b8777fcd21b429094a24c8ff-90ac0eb7-730407d6"
```

5.2. Kiểm thử