

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

**Vân Duy Quang
Lê Thái Như Quỳnh
Lê Trần Thiện Thắng
Trần Minh Toàn
Võ Duy Trường
Nguyễn Lê Công Quý**

HƯỚNG DẪN TRIỂN KHAI RELEASE

**HỆ THỐNG QUẢN LÝ KHÁCH HÀNG
CHO DOANH NGHIỆP NHỎ**

**THỰC TẬP DỰ ÁN TỐT NGHIỆP
CHƯƠNG TRÌNH CHÍNH QUY**

Thành phố Hồ Chí Minh, tháng 01/2024

Mục lục

Mục lục	i
Tóm tắt	iv
Chương 1. Cài đặt công cụ cần thiết	1
1.1. Cài đặt Docker	1
1.1.1. Tải và cài đặt Docker	1
1.1.2. Tạo tài khoản và đăng nhập DockerHub (không bắt buộc)	1
1.1.3. Kết quả	2
Chương 2. Tạo các Docker Image	3
2.1. Tạo file image	3
2.2. Đăng nhập vào Docker Hub (hoặc AWS ECR)	4
2.2.1. Docker Hub	4
2.2.2. AWS ECR	4
2.3. Tạo repository trên Docker Hub hoặc AWS ECR	5
2.4. Push Image lên Docker Hub hoặc AWS ECR	5
2.5. Lặp lại với các service khác	5

Chương 3. Release front-end	6
3.1. Tải lên thư mục	6
3.1.1. Tiến hành release	6
3.1.2. Xuất bản Release	6
Chương 4. Kết quả Biên dịch	8

Danh sách hình

1.1.1.1	Màn hình chọn hệ điều hành cho Docker	2
2.1.0.1	Docker đã tạo Image thành công	4
3.1.1.1	Mục tạo release mới	7
3.1.2.1	Tạo một phiên bản release	7

Tóm tắt

Tài liệu này được ra đời với mục đích hướng dẫn người đọc cách triển khai các kết quả biên dịch của đề tài Thực tập dự án tốt nghiệp "Hệ thống quản lý khách hàng cho doanh nghiệp nhỏ" lên một hệ thống quản lý các phiên bản release (GitHub, Docker Hub hoặc AWS ECR).

Đề tài được thực hiện theo kiến trúc microservice, do đó tất cả service hoàn toàn có thể được triển khai độc lập. Sau khi kết thúc quá trình biên dịch như được hướng dẫn ở file `CompilationGuide.pdf`, ta cần tiến hành đưa những sản phẩm sau đây lên một trong những hệ thống quản lý phiên bản đã kể trên.

1. File JAR của các service dưới đây, mỗi file đều nằm trong thư mục `salesync/back-end/service-tương-ứng/artifacts`.
 - (a) `authentication`
 - (b) `type-service`
 - (c) `record-service`
 - (d) `service-registry`
 - (e) `api-gateway`

(f) notification

2. Thư mục `dist` chứa mã nguồn HTML5 hoặc toàn bộ nội dung thư mục `salesync/front-end`.

Tất cả nội dung và sản phẩm thực hiện đồ án kể trên đều được đi kèm trong file `SourceCode.zip`. Tài liệu sẽ bao gồm hướng dẫn release cho các service kể trên trong các hệ điều hành **Windows**, **MacOS**.

Chương 1

Cài đặt công cụ cần thiết

1.1. Cài đặt Docker

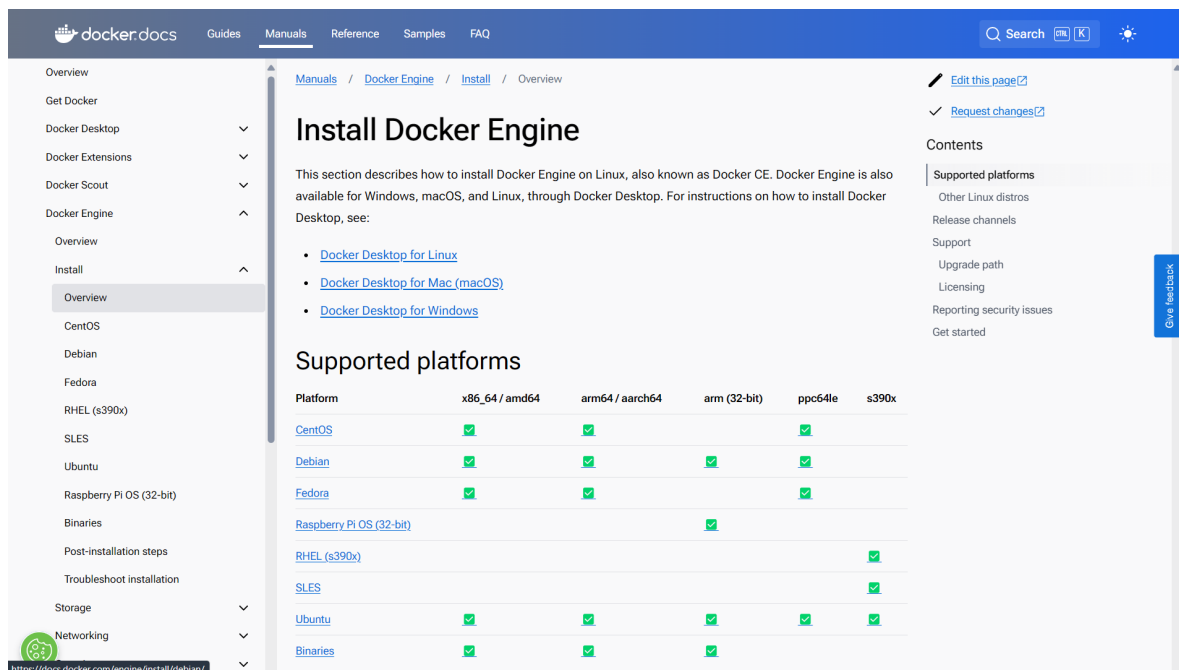
Để thuận tiện trong việc lập trình và triển khai ứng dụng, hầu hết mọi service trong đề tài đều được thực hiện thông qua Docker.

1.1.1. Tải và cài đặt Docker

Truy cập đường dẫn <https://docs.docker.com/engine/install/>. Tại màn hình này, chọn hệ điều hành như ý (Xem hình 1.1.1.1) và tiến hành tải về, cài đặt Docker với những cấu hình mặc định theo hướng dẫn riêng của từng hệ điều hành.

1.1.2. Tạo tài khoản và đăng nhập DockerHub (không bắt buộc)

Sau khi đã cài đặt thành công, ta tiến hành đăng nhập hoặc tạo tài khoản Docker để truy cập Docker Hub.



Hình 1.1.1.1: Màn hình chọn hệ điều hành cho Docker

1.1.3. Kết quả

Sau khi đã làm theo hướng dẫn cài đặt trên chủ, Docker sẽ được cài đặt và Docker Engine sẽ được chạy thành công. Để kiểm tra liệu Docker Engine đã được chạy, nhập câu lệnh `docker ps`.

Chương 2

Tạo các Docker Image

Phần này sẽ hướng dẫn tạo và release các Docker Image tương ứng cho từng service sau.

1. Service authenticaion
2. Service type-service
3. Service record-service
4. Service notification-service
5. Service service-registry
6. Service api-gateway

2.1. Tạo file image

Từ thư mục gốc salesync của mã nguồn. Truy cập vào thư mục back-end và truy cập thư mục con tương ứng với tên của service cần biên dịch. Tiến

```
\salesync\back-end\authentication> docker buildx build . -t auth:latest
[+] Building 3.2s (9/9) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 426B                             0.0s
=> [internal] load metadata for docker.io/library/openjdk:17-alpine 2.6s
=> [auth] library/openjdk:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [1/3] FROM docker.io/library/openjdk:17-alpine@sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e9dbd9 0.0s
=> [internal] load build context                                0.3s
=> => transferring context: 96B                                    0.3s
=> CACHED [2/3] WORKDIR /app                                    0.0s
=> CACHED [3/3] COPY artifacts/*.jar authentication.jar          0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:6b711311765c8185994917a3f9b6d1147e6bbe97eea25f44e5e7a75f0b9a8e8 0.0s
=> => naming to docker.io/library/auth:latest                   0.0s

View build details: docker-desktop://dashboard/build/default/default/lgl455uuzusarnhrvs4ra426l
```

Hình 2.1.0.1: Docker đã tạo Image thành công

hàng chạy lệnh

```
docker buildx build . -t tên-image:latest
```

để Docker tìm đến file Dockerfile trong từng service và tiến hành tạo ra Container Image dựa trên file JAR trong thư mục artifacts đã chuẩn bị. Ta dùng argument `-t tên-image:latest` để đặt tên cho image trên máy tính, giúp thuận tiện hơn trong việc truy vấn.

Nếu thành công, kết quả trả về sẽ có dạng như Hình 2.1.0.1.

2.2. Đăng nhập vào Docker Hub (hoặc AWS ECR)

2.2.1. Docker Hub

Đối với Docker Hub, chạy câu lệnh `docker login` và tiến hành nhập Username và password của tài khoản Docker Hub.

2.2.2. AWS ECR

Đối với AWS ECR, hướng dẫn này mặc định `aws-cli` đã được cài đặt và cấu hình trên máy tính (Xem hướng dẫn chi tiết tại đây).

Tương tự, ta cũng chạy lệnh đăng nhập đến AWS ECR, nhưng với cách thức lấy mật khẩu khác:

```
docker login -u AWS -p $(aws ecr get-login-password --region  
tên-region-aws)  
mã-container-registry.dkr.ecr.ap-southeast-2.amazonaws.com
```

2.3. Tạo repository trên Docker Hub hoặc AWS ECR

Dựa theo hướng dẫn riêng của từng container repository, ta cần tạo một repository trống để chứa image cho một service. Kết quả của quá trình này chính là một mã repository.

2.4. Push Image lên Docker Hub hoặc AWS ECR

Trước tiên, ta cần gán tag để image vừa build tại máy tính đến tên repository đã tạo.

```
docker tag tên-image:latest mã-repository:latest
```

Sau đó, chạy lệnh sau để push image lên repository. `docker push mã-repository`

Chú ý: Trong những bước đã liệt kê, có thể thay đổi tag latest thành tên của số hiệu phiên bản mong muốn.

2.5. Lặp lại với các service khác

Thực hiện những bước tương tự với các service khác, kết quả đạt được bao gồm 6 repository khác nhau tương ứng với 6 service đã liệt kê. Đây cũng là sản phẩm cuối cùng của quy trình tạo và release image.

Chương 3

Release front-end

Để release các phiên bản khác nhau của front-end, ta chỉ cần tải lên toàn bộ thư mục lên một repository GitHub trống. Cài đặt này mặc định đã tồn tại một tài khoản GitHub và một repository trống.

3.1. Tải lên thư mục

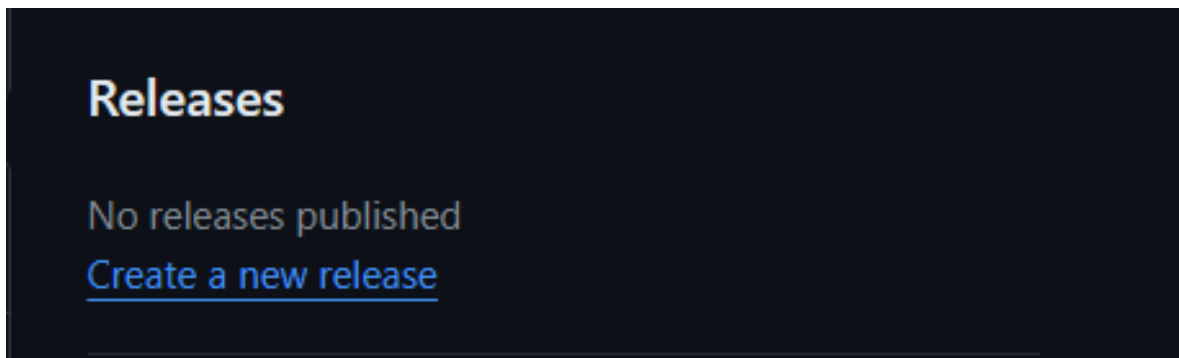
Tại repository trống, tiến hành tải lên toàn bộ thư mục.

3.1.1. Tiến hành release

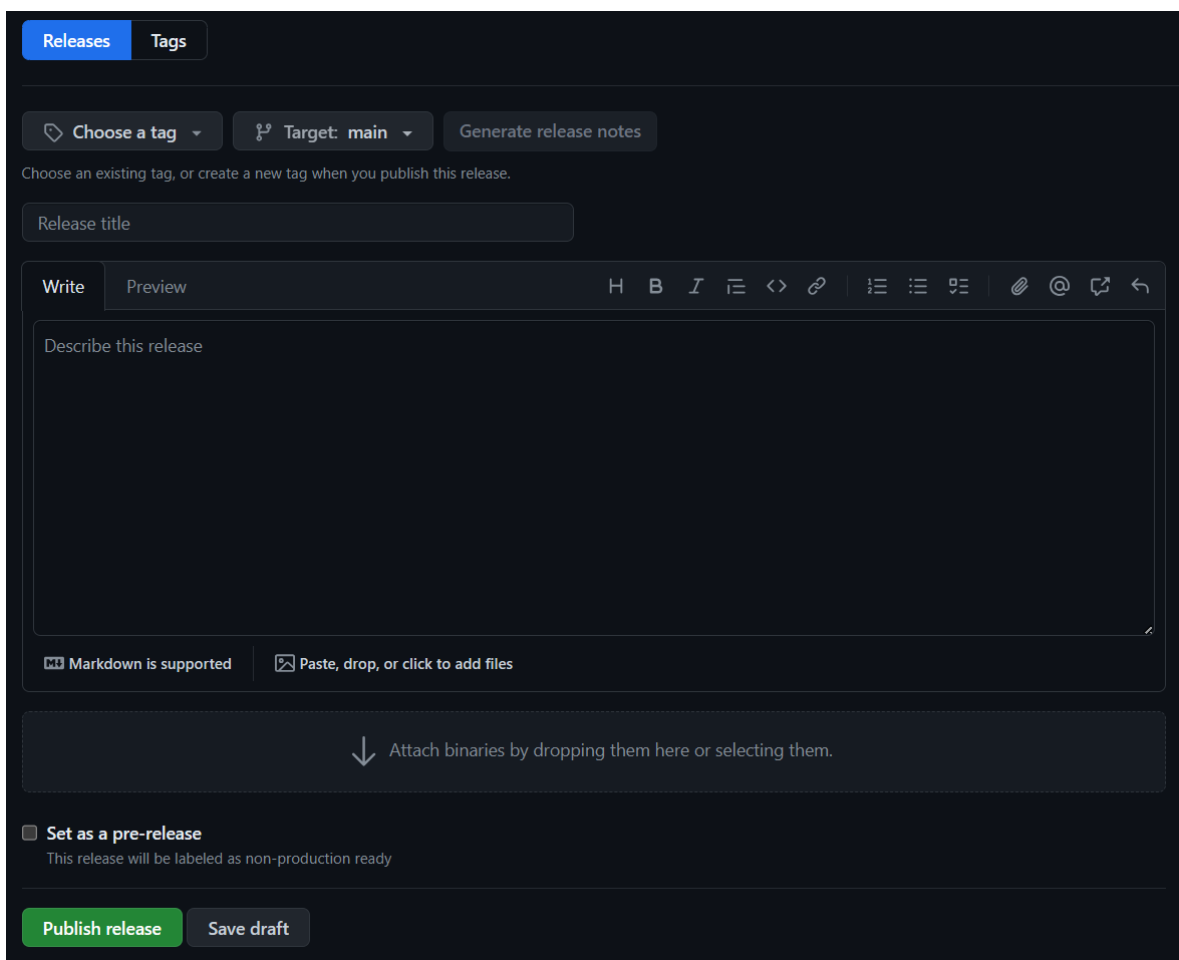
Tại trang chủ của repository, chọn `Create a new release` tại mục `Release` (Xem Hình 3.1.1.1).

3.1.2. Xuất bản Release

Điền tất cả những thông tin cần thiết cho bản release và chọn `Publish Release`. (Xem Hình 3.1.2.1)



Hình 3.1.1.1: Mục tạo release mới



Hình 3.1.2.1: Tạo một phiên bản release

Chương 4

Kết quả Biên dịch

Sau khi kết thúc quá trình release các service back-end và front-end. Kết quả thu được sẽ bao gồm.

1. 6 repository tương ứng với 6 service đã liệt kê.
2. 1 GitHub repository dành riêng cho front-end.

Tài liệu DeploymentGuide.pdf đi kèm sẽ tiếp tục hướng dẫn cách sử dụng các kết quả release này để phục vụ cho việc triển khai toàn bộ hệ thống lên các máy chủ.