RabbitMQ

1. Tại sao lại có **RabbitMQ**?

- Trong TH một service nhận quá nhiều request mỗi giây, và chúng có thể bị blocked các request sau đó để request trước đó xử lý.

- Vì vậy chúng ta sẽ đặt các request đó vào một queue giữa web service và processing serivce. Lúc này 2 service sẽ hoàn toàn tách rời nhau.

- Hơn nữa các request của chúng ta đều được lưu lại và được xử lý lần lượt

2. Một số khái niệm:

- Producer: Ứng dụng gửi message

- Consumer: Ứng dụng nhận message

- Queue: Lưu trữ message

- Connection: Một kết nối TCP giữa ứng dụng RabbitMQ brocker

- Channel: Một kết nối ảo trong một connection, việc publishing hoặc consuming từ một queue đều được thực hiện trên channel.

- Exchange: Là nơi nhận message được publish từ Producer và đẩy chúng vào queue dựa trên nguyên tắc của từng Exchange. Để nhận được message, queue phải được nằm trong ít nhất một Exchange.

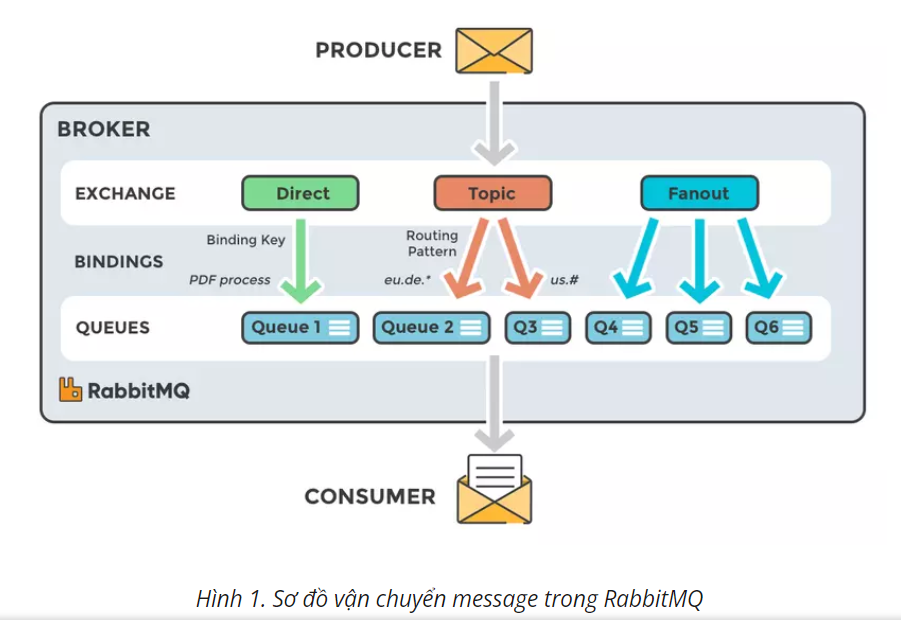
- Binding: Đảm bảo nhiệm vụ liên kết giữa Exchange và Queue.

- Routing key: Một key mà Exchange dựa vào đó để quyết định cách để định tuyến message đến Queue. Có thể hiểu nôm na, Routing key là địa chỉ dành cho message.

- AMQP: Giao thức Advance message Queuing Protoco, là giao thức truyền message trong RabbitMQ.

- User: Để có thể truy cập vào RabbitMQ, chúng ta phải có username và password. Trong RabbitMQ, mỗi user được chỉ định với một quyền hạn nào đó. User có thể phân quyền đặc biệt trong một Vhost nào đó.

- Virtual host/Vhost: Cung cấp những cách riêng biệt để các ứng dụng có thể dụng chung một RabbitMQ instant. Những user khác nhau có thể có các quyền khác nhau đới với Vhost khác nhau. Queue và exchange có thể được tạo, vì vậy chúng chỉ tồn tại trong một Vhost.



3. Các loại exchange

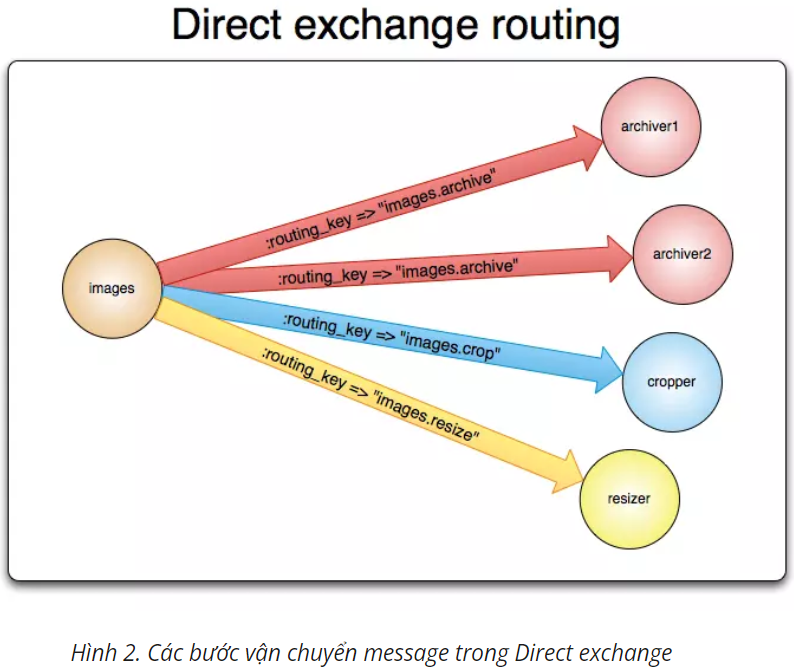
a. Direct exchange:

Vận chuyển message tới queue dựa vào routing key. Thường được xử dụng cho việc định tuyến tin nhắn unicase-đơn hướng (mặc dù nó có thể xử dụng cho định hướng multicase-đa hướng).

Các bước định tuyến message:

+) Một queue được ràng buộc với một direct exchange bởi một routing key K.

+) Khi có một message mới với routing key R đến direct exchange, message sẽ được chuyển tới queue đó nếu R=K.



b. Default exchange:

- Mỗi một exchange đều được đặt một tên không trùng nhau, default exchange bản chất là một direct exchange nhưng không có tên(String rỗng). Nó có thuộc tính đặc biệt làm cho nó rất hữu ích cho các ứng dụng đơn giản: mọi queue được tạo sẽ tự động được liên kết với nó bằng một routing key giống như tên queue.

- Ví dụ, nếu bạn muốn tạo ra một queue với tên “hello-word”, RabbitMQ broker sẽ tự đọng binding default exchange đến queue “hello-word” với routing key “hello-word”.

C. Topic exchange

- Topic exchange định tuyến message tới một hoặc nhiều queue dựa trên sự trùng khớp giữa routing key và pattern. Topic exchange thường sử dụng để thực hiện định tuyến thông điệp multicast.

- Ví dụ TH sử dụng:

+ Phâm phối dữ liệu liên quan đến vị trí địa lý cụ thể

+ Xử lý tác vụ nền được thực hiện bởi nhiều workers, mỗi công việc có khả năng xử lý các nhóm tác vụ cụ thể.

+ Cập nhật tin tức liên quan đến phân loại hoặc gắn thể (Ví dụ: chỉ dành cho một môn thể thao hoặc đội cụ thể)

+ Điều phối các dịch vụ của các loại khác nhau trong cloud

D. Headers exchange

- Header exchange được thiết kế để định tuyến với nhiều thuộc tính, dễ dàng thực hiện dưới dạng tiêu đề của message hơn là routing key. Header exchange bỏ đi routing key mà thay vào đó định tuyến dựa trên header của message. Trường hợp này, broker cần một hoặc nhiều thông tin từ application developer, cụ thể là, nên quan tâm đến những tin nhắn với tiêu đề nào phù hợp hoặc tất cả chúng.