

주요 용어와 Exchange의 이해

RabbitMQ 주요 용어 정리

1. Producer (생산자):

- 메시지를 생성하고 RabbitMQ에 전송하는 애플리케이션
- Producer는 특정 Exchange에 메시지를 전송하고 Exchange는 메시지를 라우팅하여 큐에 배치

2. Exchange :

- Producer로부터 받은 메시지를 큐에 전달
- Exchange 유형:
 - Direct: 특정 라우팅 키와 정확히 일치하는 큐에 메시지를 전송
 - Fanout: 모든 큐에 메시지를 브로드캐스트
 - Topic: 라우팅 키 패턴을 기반으로 메시지를 특정 큐에 전달
 - Headers: 메시지 헤더 속성에 따라 메시지를 라우팅
- 메시지가 Exchange로 전송될 때, Routing Key가 함께 전달

3. Routing Key :

- 메시지를 전송할 때 Producer가 Exchange에 전달하는 키
- Exchange는 이 Routing Key를 참고하여 어떤 큐에 메시지를 전달할지 결정

4. Queue :

- 메시지를 일시적으로 저장하는 버퍼 역할
RabbitMQ의 큐는 FIFO(First In, First Out) 방식으로 동작하며, 메시지가 소비자에게 전달될 때까지 보관
- 각 큐는 여러 Consumer가 구독(수신)할 수 있으며, 메시지는 큐에 들어온 순서대로 전달
- 비동기적으로 동작하며, 여러 컨슈머가 동시에 메시지를 소비할 수 있다. 단 하나의 메시지가 여러 소비자에게 중복으로 전달될수는 없음
동일한 메시지를 수신하려면 Fanout Exchange 방식으로 동작해야만 함.

5. Binding :

- exchange와 큐간의 관계를 정의
- 바인딩은 메시지를 라우팅할 때 어떤 조건으로 큐에 보낼지 정의하고 이를 위해 binding key가 사용됨
- Binding Key와 Routing Key가 일치하면 해당 큐로 메시지가 전달 (패턴 매칭 가능)

6. Consumer (소비자):

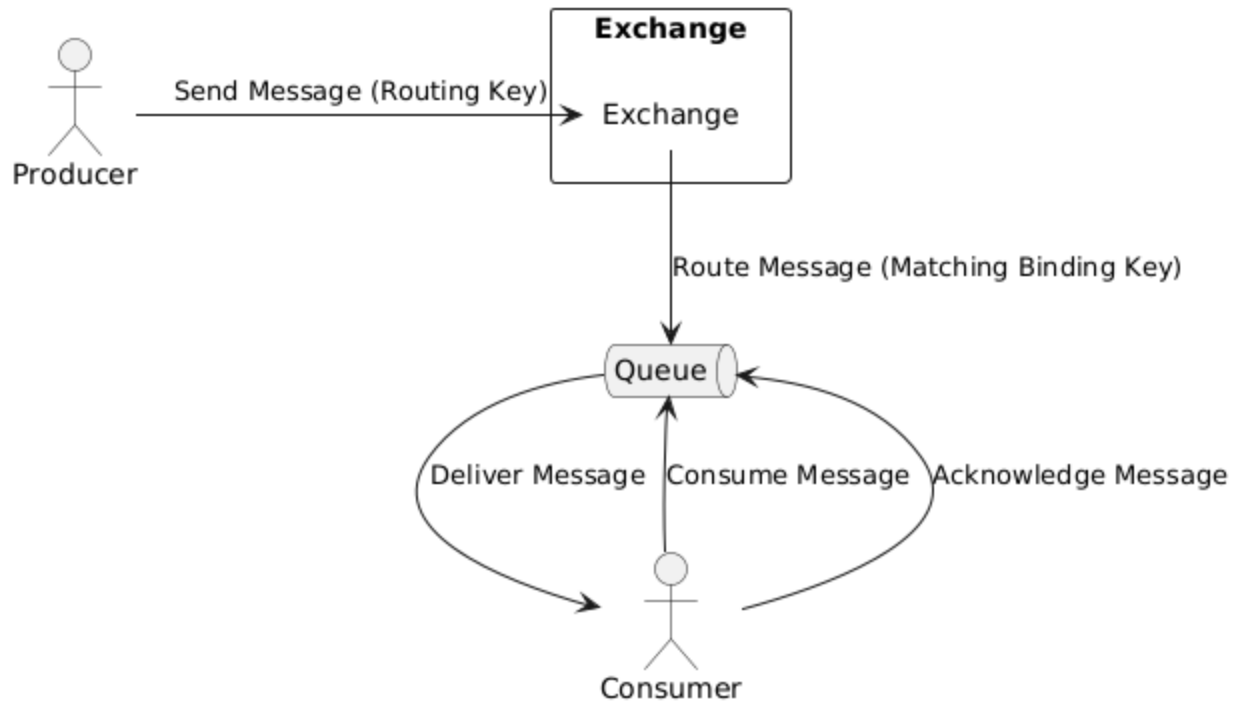
- 큐에서 메시지를 가져와 처리하는 애플리케이션
- RabbitMQ는 여러 소비자에게 메시지를 로드 밸런싱 할 수 있다.
- Consumer는 큐에서 메시지를 받아 처리하면 메시지에 대한 확인(ACK, acknowledgment)을 브로커에 전송함
- 확인을 보내지 않으면, 브로커는 메시지를 재전송하거나 설정한 다른 Consumer에게 전달할 수 있다.

7. Message Acknowledgment (메시지 확인):

- 메시지가 성공적으로 처리되었음을 RabbitMQ에 알리는 과정
- 만약 소비자가 메시지를 성공적으로 처리하지 못했다면, 메시지를 다시 큐에 넣어 다른 소비자가 처리하도록 할 수 있다.

이 과정을 간략히 도식화 하면 아래와 같다.

1. Producer가 메시지와 Routing Key를 Exchange에 전송.
2. Exchange가 Routing Key를 사용해 Binding Key가 일치하는 큐에 메시지를 라우팅.
3. Consumer가 큐에서 메시지를 가져와 처리하고, 성공적으로 처리되었음을 acknowledgment로 RabbitMQ에 알림.



추가로 알아야 할 용어

1. Prefetch Count (프리페치 카운트):

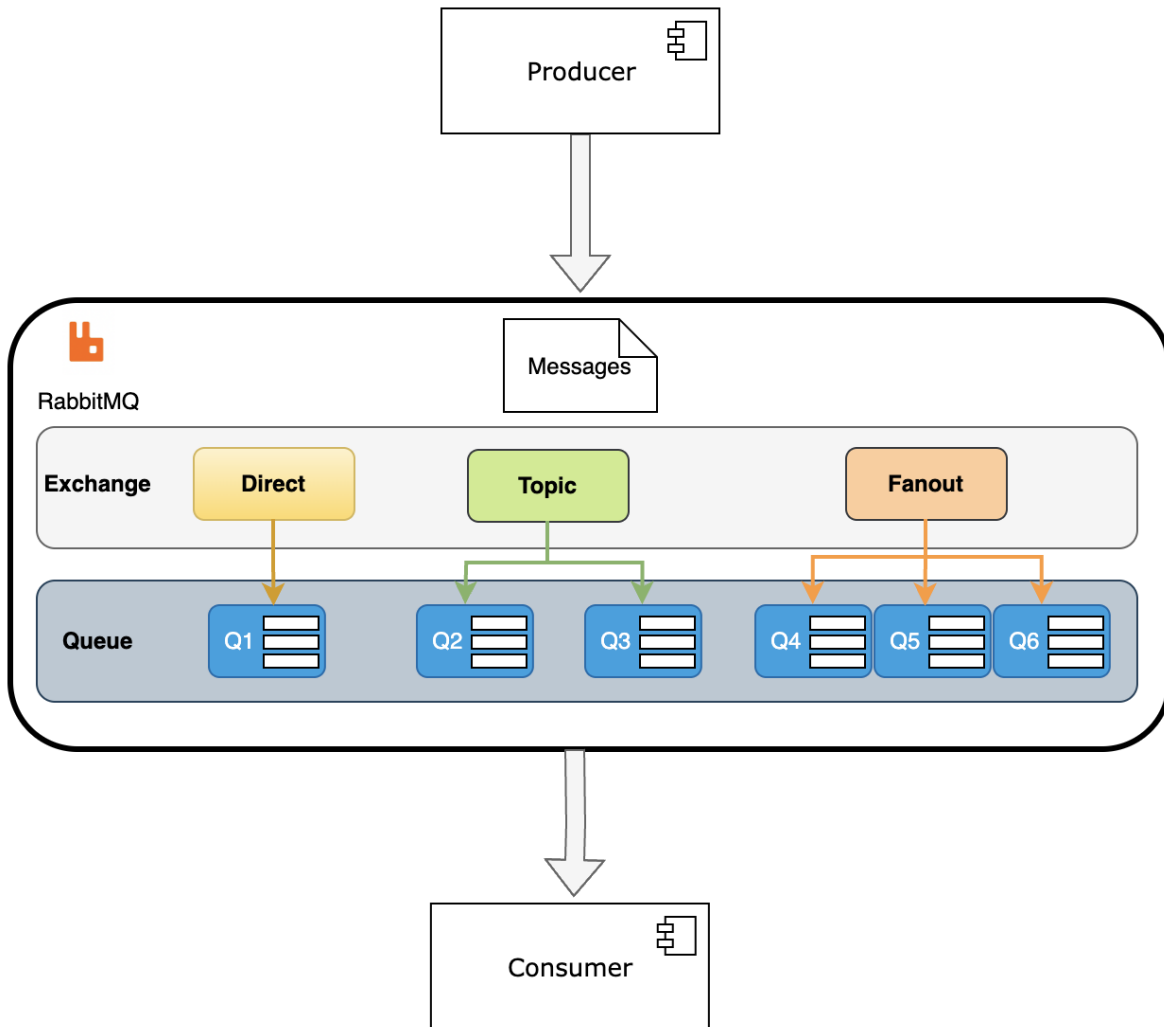
- 소비자가 받을 수 있는 최대 메시지 수를 설정
- 한 번에 많은 양의 메시지를 처리하지 않도록 하여 소비자의 성능 최적화

2. Virtual Host (가상 호스트):

- RabbitMQ 서버 내의 논리적인 구획으로, 메시지 큐, 익스체인지, 사용자 권한 등을 구분
- 하나의 RabbitMQ 서버 내에 여러 개의 가상 호스트를 설정하여 서로 다른 애플리케이션의 메시지를 격리 할 수 있다.

3. Dead Letter Queue (DLQ) : 메시지가 처리되지 못하거나 유효 기간이 지난 경우 별도의 큐로 이동하는 구조도 설정할 수 있다.

Exchange 유형에 따른 처리 흐름



1. Direct Exchange

Direct Exchange는 메시지가 라우팅 키(Routing Key)에 따라 특정 큐로 하나씩 전달되는 방식이다. 메시지를 발행할 때 사용하는 라우팅 키와 동일한 키로 익스체인지에 바인딩 된 모든 큐에 메시지를 전달한다. 해당 라우팅 키와 일치하는 큐에만 메시지가 전달되는 방식이기 때문에 Direct Exchange 라고 한다.

사용 예시: 주문에 따른 상태 지시

활용: 주문 상태별로 **라우팅 키**를 정의하고, 각 상태에 해당하는 큐가 메시지를 받는다. 매핑이 정확하게 되는 한개의 키만 있으니까 1:1로 가능할거 같은데, **하나의 라우팅 키**에 대해 **여러 큐**가 바인딩될 수 있기 때문에 **1:N** 매칭이 가능하다.

- 메시지가 **명확하게 특정 큐로 전달**되어야 할 때.
- 큐마다 **고유한 라우팅 규칙**을 적용하여 메시지를 분류해야 할 때.
- **예시 업무**: 주문 상태 처리, 결제 처리, 사용자 알림 시스템 등.

2. Topic Exchange:

Topic Exchange는 라우팅 키를 **패턴 기반**으로 정의하여 메시지를 여러 큐에 유연하게 전달할 수 있는 방식이다. 라우팅 키에 와일드카드(*, #) 매칭을 사용하여 더 복잡한 라우팅이 가능하다.

- 와일드카드 * 의 경우 하나의 단어를 대체 하는 의미로 log.info, log.warn, log.error 와 같은 패턴의 메시지를 수신 할 때 log.* 로 info와 warn, error 를 다 수신하게 만들수 있다.
- #의 경우 0개 이상의 단어를 대체하므로 app.order.success, app.payment.success 와 같은 라우팅 키를 #.success 로 다 수신할 수 있다.

사용 예시: 동적이고 유연한 라우팅이 필요할 때(로그 수집 시스템, 이벤트 기반 모니터링 등)

3. Fanout Exchange :

Fanout Exchange는 **브로드캐스트** 방식으로 메시지를 모든 바인딩된 큐에 전달한다. 한 번의 메시지 발행으로 모든 큐가 동일한 메시지를 받는다.

사용 예시: 이벤트가 발생하면 모든 서비스가 동일한 메시지를 받는 서비스에서 유용하다. (시스템 점검 공지 등)

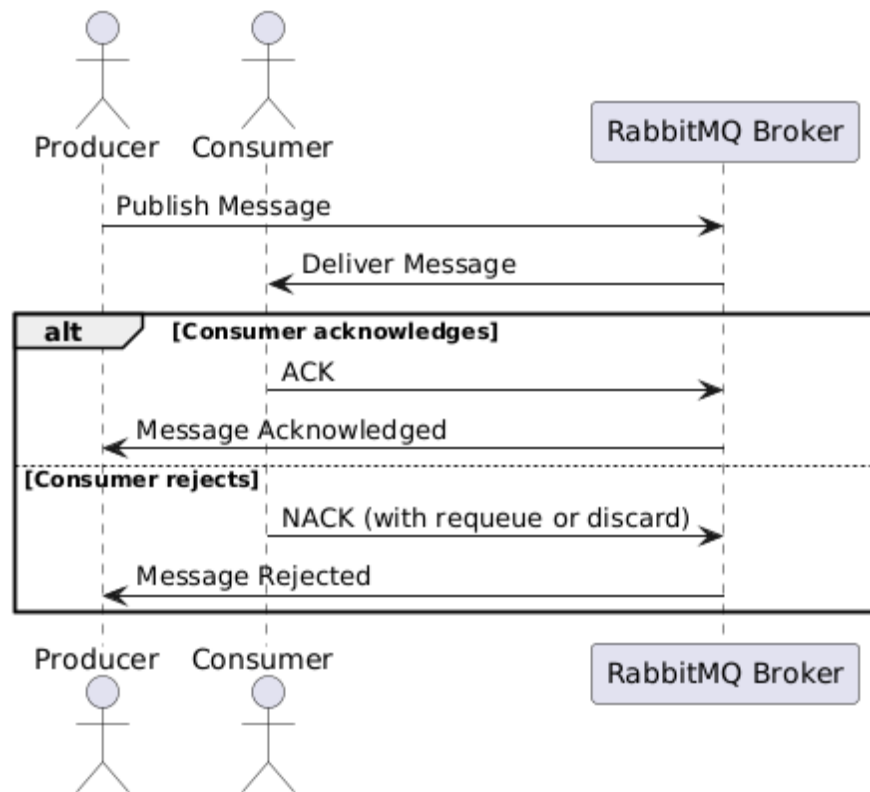
4. **Headers Exchange**: 메시지의 속성(헤더)에 기반한 복잡한 라우팅이 필요할 때.

사용 예시: 다국어 서비스, 고객의 등급별 혜택 알림.

메시지 헤더에 language: "ko", language: "en" 등의 값을 설정하여 헤더 기반 라우팅을 수행

- language: "ko"로 설정된 메시지는 한국어 이메일 서비스에서 처리.
- language: "en"으로 설정된 메시지는 영어 이메일 서비스에서 처리.

메시지 전송 단계별 프로세스



1. 메시지 송신 (Producer → Broker)

- **Producer**가 **RabbitMQ Broker**로 메시지를 송신
이때 메시지는 큐에 저장되며, 익스체인지와 바인딩 설정에 따라 적절한 큐로 라우팅

2. 메시지 전달 (Broker → Consumer):

- **Broker**는 큐에 있는 메시지를 **Consumer**에게 전달.
Consumer는
큐에서 메시지를 가져가거나(**Polling**) 메시지를 푸시(**Push**) 받는 방식으로 수신

3. 메시지 확인(ACK) 또는 거절(NACK):

- **ACK**: **Consumer**가 메시지를 성공적으로 처리한 후 **Broker**에 **ACK(Acknowledgment)**를 전송.
이 경우 **Broker**는 해당 메시지를 큐에서 제거하고 **Producer**에게 **Message Acknowledged** 응답을 전송

- **NACK**: Consumer가 메시지 처리에 실패하거나 메시지를 거절할 경우 NACK(Negative Acknowledgment)을 전송.

NACK에는 메시지를 다시 큐로 보내야 할지(requeue) 또는 폐기해야 할지(discard) 설정 가능

재전송 요청 (Requeue): 메시지를 다시 큐로 보내고 재처리할 수 있도록 설정

폐기(Discard): 메시지를 큐에서 제거하고, 폐기 처리

- Consumer가 메시지를 NACK하면 Broker는 Producer에게 **Message Rejected** 응답을 전송

4. **Producer에 응답 (Message Acknowledged / Message Rejected):**

- Producer가 **Publisher Confirms**를 활성화한 경우, Broker는 ACK 또는 NACK 결과를 Producer에게 전송
- ACK를 받은 경우 메시지가 성공적으로 소비된 것으로 간주되며, NACK를 받은 경우 Producer는 메시지 실패를 기록하거나 재전송