

Pub/Sub을 이용해 손쉽게 채팅방 기능 구현하기

1 Pub/Sub 패턴의 이해

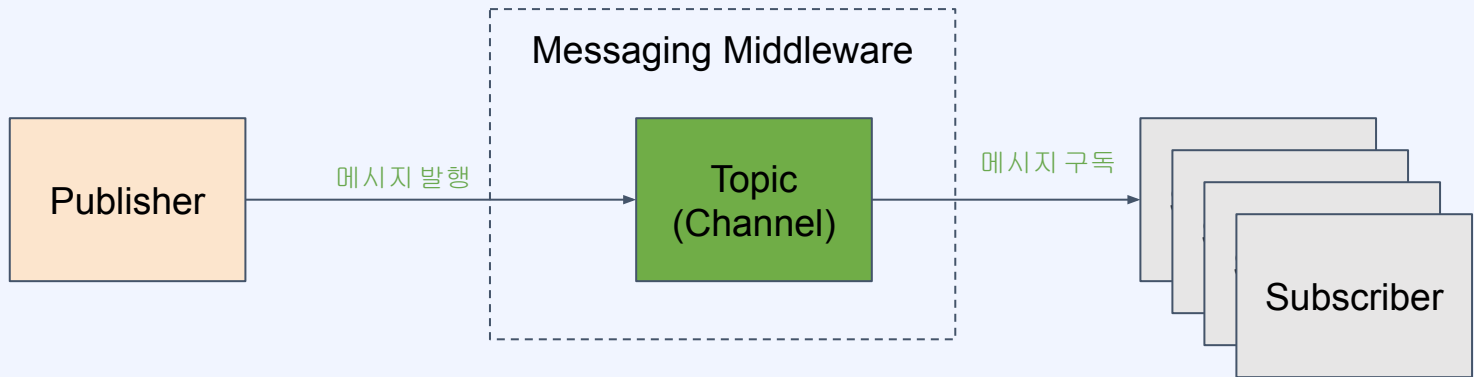
Pub/Sub 패턴이란?

1.

Pub/Sub 패턴의
이해

Pub/Sub 패턴

- 메시징 모델 중의 하나로 발행(Publish)과 구독(Subscribe) 역할로 개념화 한 형태
- 발행자와 구독자는 서로에 대한 정보 없이 특정 주제(토픽 or 채널)를 매개로 송수신



Pub/Sub 패턴이란?

1.

Pub/Sub 패턴의
이해

메시징 미들웨어 사용의 장점

- 비동기: 통신의 비동기 처리
- 낮은 결합도: 송신자와 수신자가 직접 서로 의존하지 않고 공통 미들웨어에 의존
- 탄력성: 구성원들간에 느슨한 연결로 인해 일부 장애가 생겨도 영향이 최소화됨

메시징 미들웨어 제품들: Kafka, RabbitMQ, ActiveMQ, ...

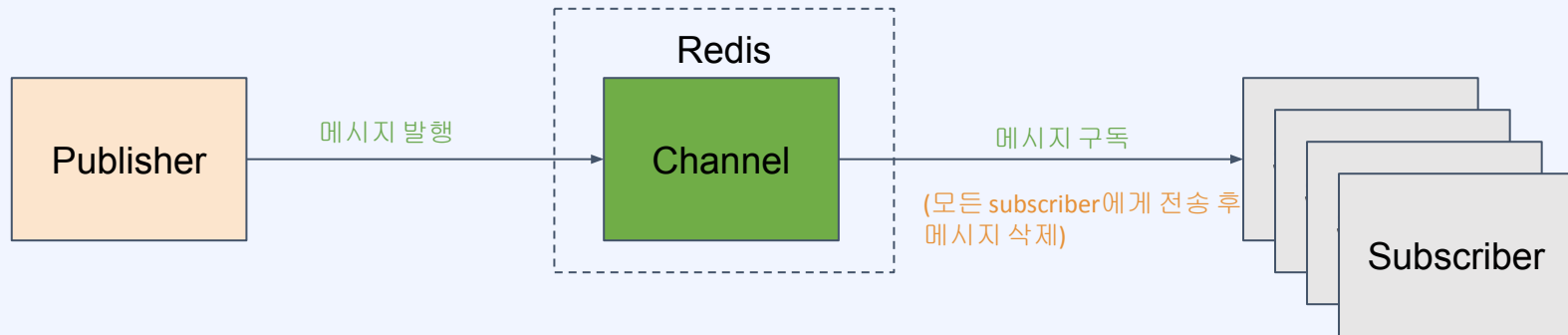
Redis의 Pub/Sub 기능

1.

Pub/Sub 패턴의
이해

Redis의 Pub/Sub 특징

- 메시지가 큐에 저장되지 않음.
- **Kafka**의 컨슈머 그룹같은 분산처리 개념이 없음.
- 메시지 발행 시 **push** 방식으로 **subscriber**들에게 전송
- **subscriber**가 늘어날수록 성능이 저하



Redis의 Pub/Sub 기능

1.

Pub/Sub 패턴의
이해

Redis의 Pub/Sub의 유즈케이스

- 실시간으로 빠르게 전송되어야 하는 메시지
- 메시지 유실을 감내할 수 있는 케이스
- 최대 1회 전송(at-most-once) 패턴이 적합한 경우
- Subscriber들이 다양한 채널을 유동적으로 바꾸면서 한시적으로 구독하는 경우

Pub/Sub을 이용해 손쉽게 채팅방 기능 구현하기

2 Redis Pub/Sub을 이용한 채팅방 구현

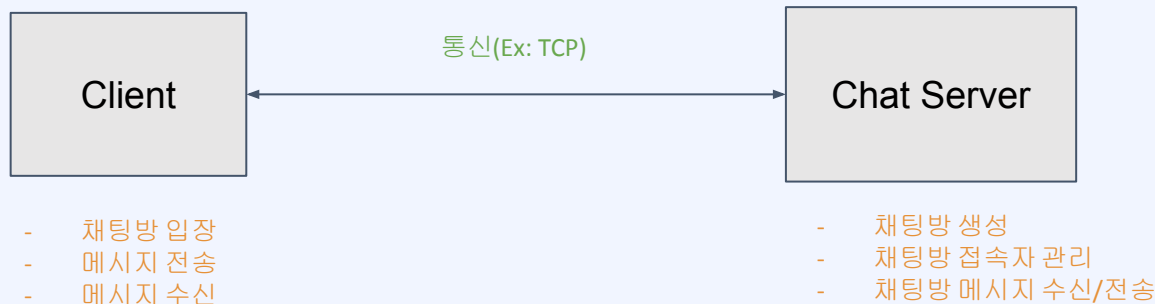
Redis의 Pub/Sub을 이용한 채팅방 구현

2.

Redis Pub/Sub을
이용한 채팅방
구현

채팅방 기능의 요구사항

- 채팅 클라이언트와 채팅 서버가 존재하고 통신 방식을 정해야 함.(프로토콜)
- 채팅 서버는 채팅방 관리 로직을 작성해야 함



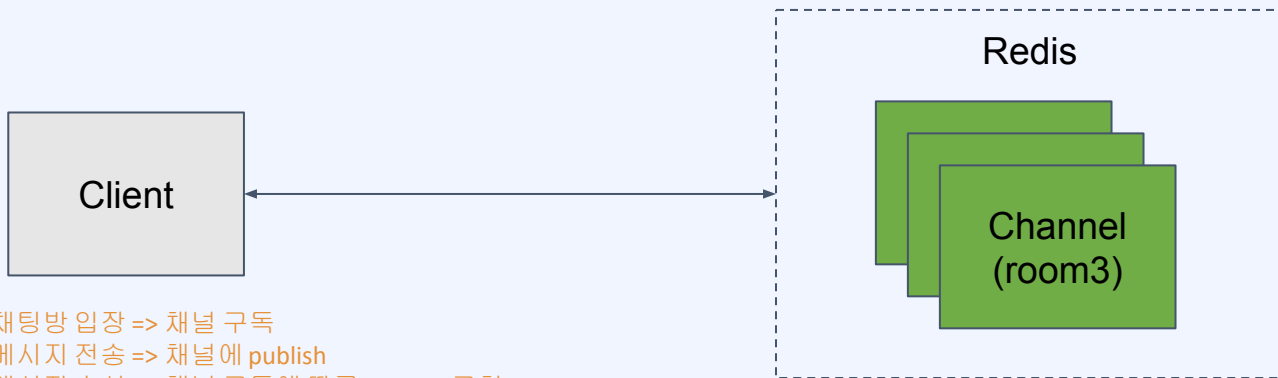
Redis의 Pub/Sub을 이용한 채팅방 구현

2.

Redis Pub/Sub을
이용한 채팅방
구현

Redis Pub/Sub을 이용한 채팅방 구현

- 채팅방 기능을 publish/subscribe 구조를 이용해 쉽게 구현



- 채팅방 입장 => 채널 구독
- 메시지 전송 => 채널에 **publish**
- 메시지 수신 => 채널 구독에 따른 **listener** 구현

- 채팅방 생성 => 채널 사용으로 대체
- 채팅방 접속자 관리 => 필요 없음
- 채팅방 메시지 수신/전송 => 필요 없음