

Developer Portfolio

- 유지보수하기 좋은 코드를 목표로 합니다.
- 협업과 자기 주도 학습을 통해
‘지속가능한 개발자’가 되기 위해 노력합니다.



황진하

hjh70902000@naver.com

github.com/madcom96

CONTENTS

- 1) Education, Certificates, Award ... 3
- 2) Skills ... 4-5
- 3) Projects Summary ... 6
- 4) Project Details ... 7-18

Education

삼성 청년 SW 아카데미 (SSAFY)	2023.07 ~ 2024.06
부산대학교 정보컴퓨터공학전공 졸업	2021.03 ~ 2023.08
한국해양대학교 제어계측전공 중퇴(편입)	2016.03 ~ 2021.02
Richard King High School 고등학교 교환학생	2013.01 ~ 2013.12

Certificates

OPIc (English) IH	2023.03
리눅스마스터 2급	2023.12
SQL개발자 (SQLD 자격)	2023.12

Award

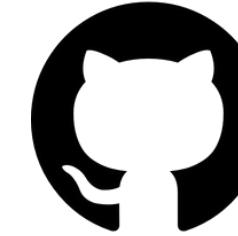
패스액 메이커톤 - NVIDIA 스폰서 트랙상	2023.02
---------------------------	----------------

Skills



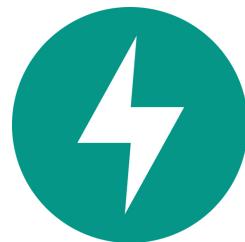
Python

알고리즘 문제풀이 (백준 상위 4%)



GitHub, GitLab

Issue, PR등을 코드리뷰에 적극적으로 활용



FastAPI

자세 추정 AI를 적용한 서버 개발
Kafka를 통한 비동기 api 콜 구현



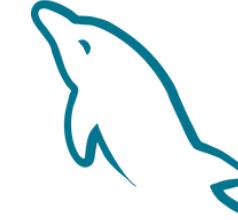
MongoDB

NoSQL 특징 이해, 활용 가능



JAVA

알고리즘 문제풀이 (백준 상위 4%)
SOLID 고려 및 디자인패턴 적용 경험



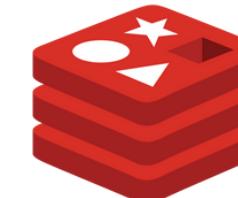
MySQL, MariaDB

트랜잭션 레벨 이해
분산 락 적용 동시성처리 경험



Spring

6인 6주 규모 프로젝트 3회 수행
RestApi 구성 가능
Spring Data JPA, Mybatis-Spring DB 연동



Redis

pub/sub 활용 분산락 적용
Redis 지원 자료구조 이해
jwt 로그아웃 처리

Skills



AWS EC2 / S3버킷

인스턴스 설정, 서버 설정, SSL 적용 경험



Jenkins

배포 스크립트 작성 가능
Blue-Green 배포 경험



Jira, Notion, Slack

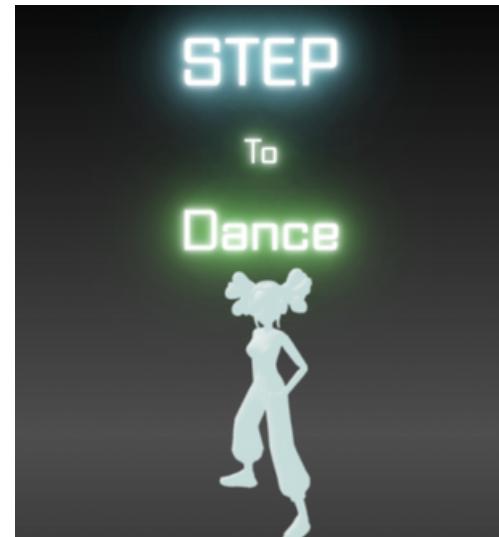
협업 및 문서화 툴을 이용하여
애자일 프로젝트 3회 수행



Docker

Dockerfile 작성 및 이미지 생성 가능
Docker Hub를 이용한 원격 배포 경험
Docker Compose 사용 경험

Projects Summary



1. Step To Dance



2024.04 ~ 2024.05 (6주)

자세 추정 AI기반
숏폼 춤 피드백 어플

기여도

Infra 100% Back-end 100%



2. Trip-Together

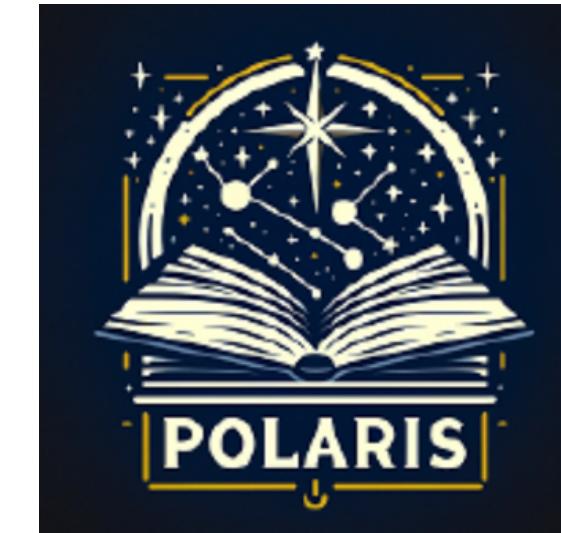


2024.02 ~ 2024.04 (6주)

여행 계획, 번개모임, 정산까지
해외여행 어플

기여도

Infra 50% Back-end 40%



3. Polaris



2024.01 ~ 2024.02 (6주)

책과 경험을 함께 공유하는
중고책 거래 플랫폼

기여도

Infra 100% Back-end 40%

Step To Dance

기간 2024.04.08 ~ 2024.05.17

팀 구성 Back-end 1인
Front-end 3인

기여도 Back-end 100%
Infra 100%

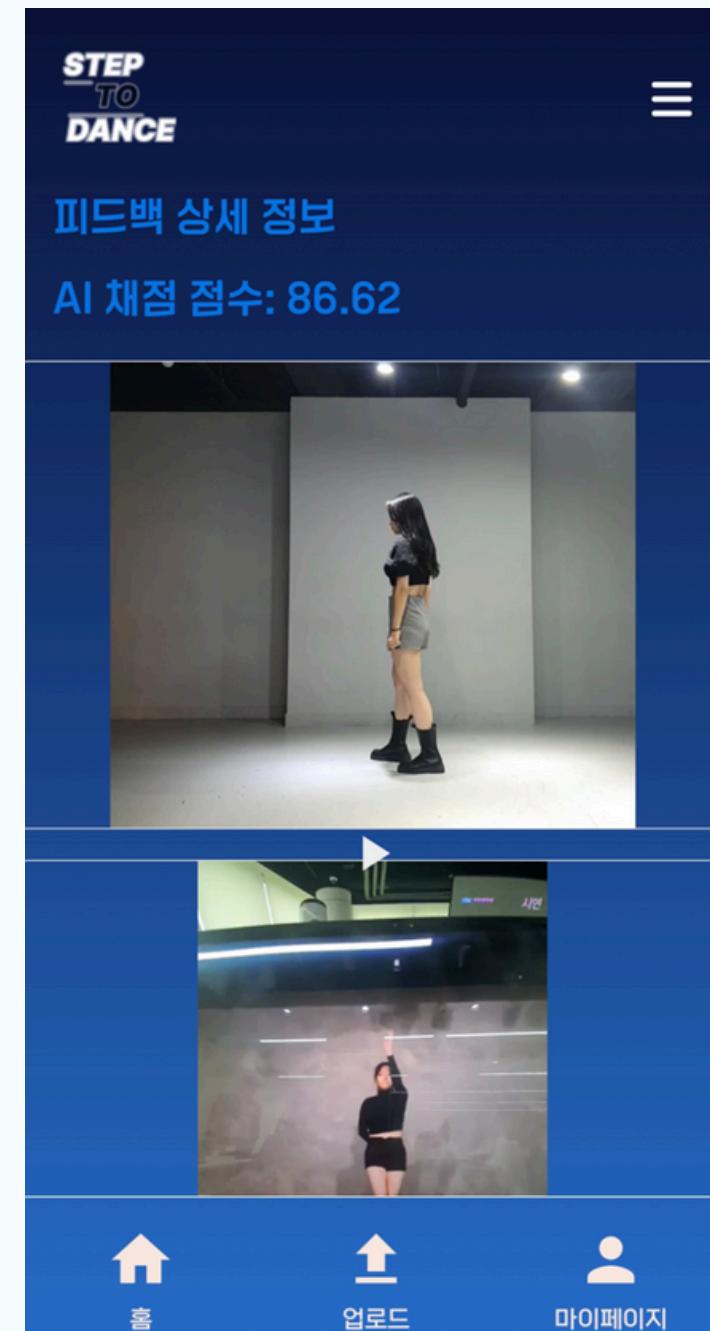
- AI의 원활한 사용을 위해 Python 서버 구성
FastAPI 프레임워크 사용
- 가이드 춤 기능과 연습 춤 기능 제공
연습 춤은 가이드 춤과의 유사도를 점수화
- 많이 선택된 가이드 춤을 'Top 5'로 선정
- 연습 춤 점수의 총합에 따른 랭킹 기능
- SNS 공유 기능

서버 Java, Spring, Python, FastAPI

기술 AI OpenPose, MoveNet

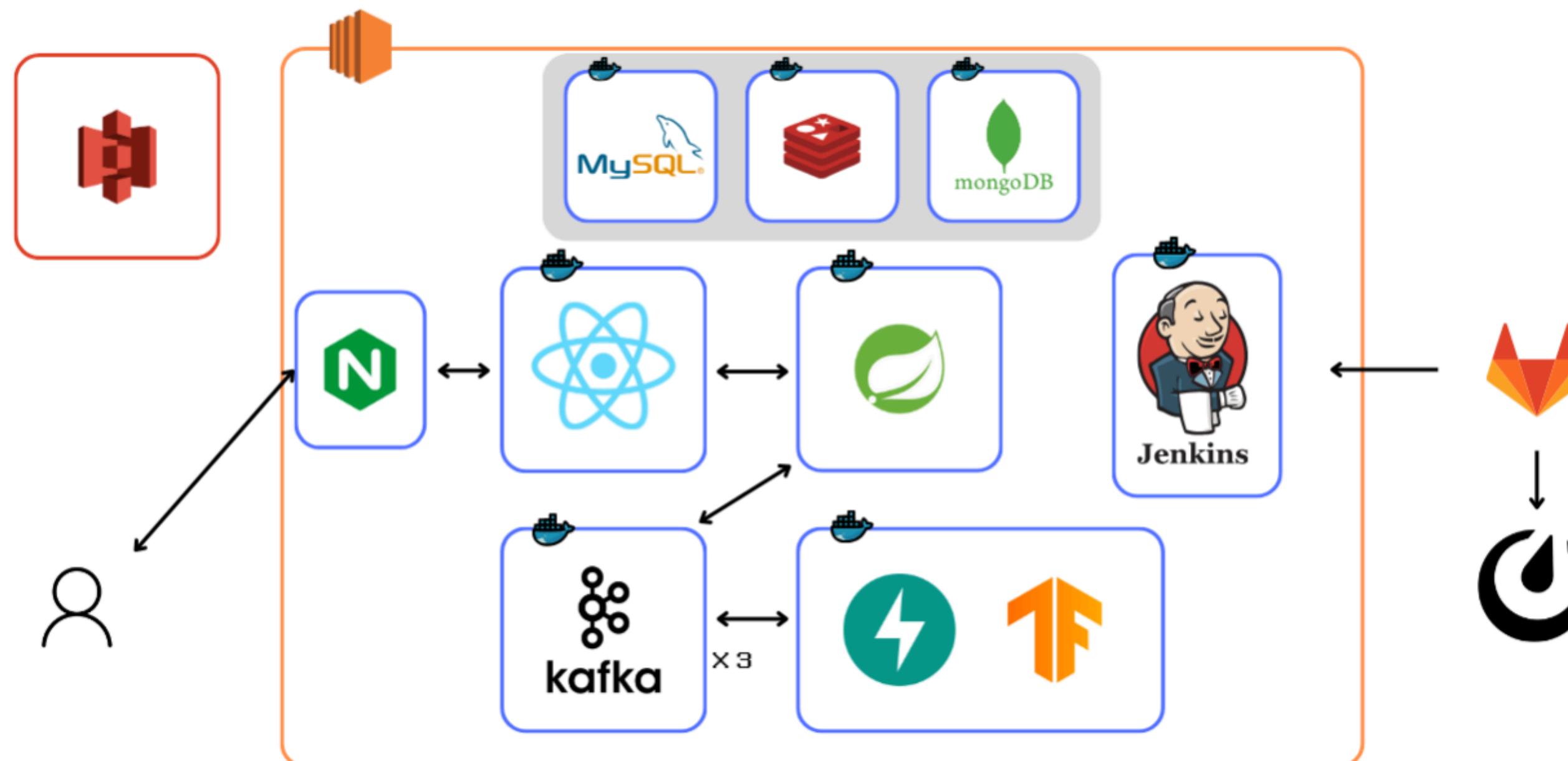
인프라 Jenkins, GitLab, Docker

소스코드 [Step To Dance](#)



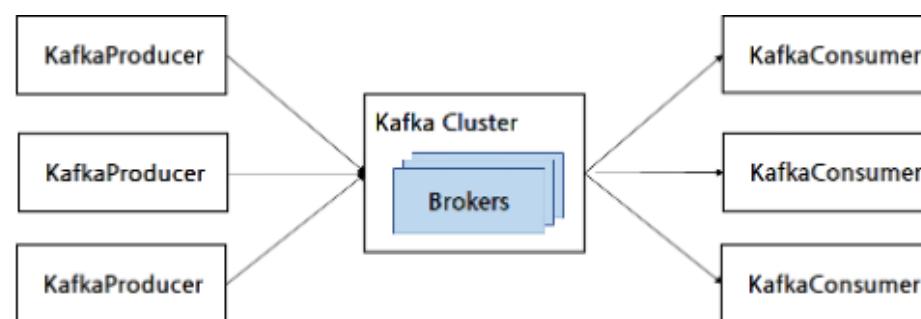
Step To Dance

System Architecture



Step To Dance

Issue : 대용량 데이터 전송



Problem : 분리된 서버간 많은 데이터 교환으로 인한 타임아웃 발생

- 서비스 서버와 AI 서버를 분리하는 과정에서 **많은 데이터 전송 발생**
- 1분 동영상의 경우 30프레임의 이미지 전송과 1프레임당 32개의 float형 배열이 프레임마다 필요

Solution : kafka를 활용한 비동기 API 콜

- Kafka 실행 오버헤드를 줄이기 위해 최신 **kraft 프로토콜 적용**
 - 기존 레퍼런스가 많던 zookeeper보다 참고자료는 적음
 - 실행 시 컴퓨팅 리소스의 효율적인 사용을 위해 kraft 선택 & 학습
- 서비스 서버와 AI 서버에 kafka 메시지 발생 시 로직 개발
 - **동영상은** 프레임마다 **이미지로 분할**하여 전송
 - AI 자세 분석 결과 데이터를 후처리하여 점수화
- 하나의 영상에서 나온 **데이터는 한번에 처리**
 - 모든 프레임에 대한 작업 완료 시, 플래그 전송
 - 플래그를 감지한 서비스 서버에서 데이터 한번에 처리

Result

- 무거운 AI모델과 가벼운 모델을 혼합하여 **안정적인 성능**으로 분석 서비스 제공
- 데이터 유실에 대한 **안전성 확보**

Step To Dance

Issue : 여러가지 AI 모델 적용



Problem : 다양한 AI 모델의 필요성

- 시간 효율, 정확성에 따른 **여러 AI 모델을 테스트하고 적용**하여, 춤 장르에 따라 적용
 - MoveNet, Openpose 사용
- 추후 개발될 AI 모델에 대한 **확장성** 필요

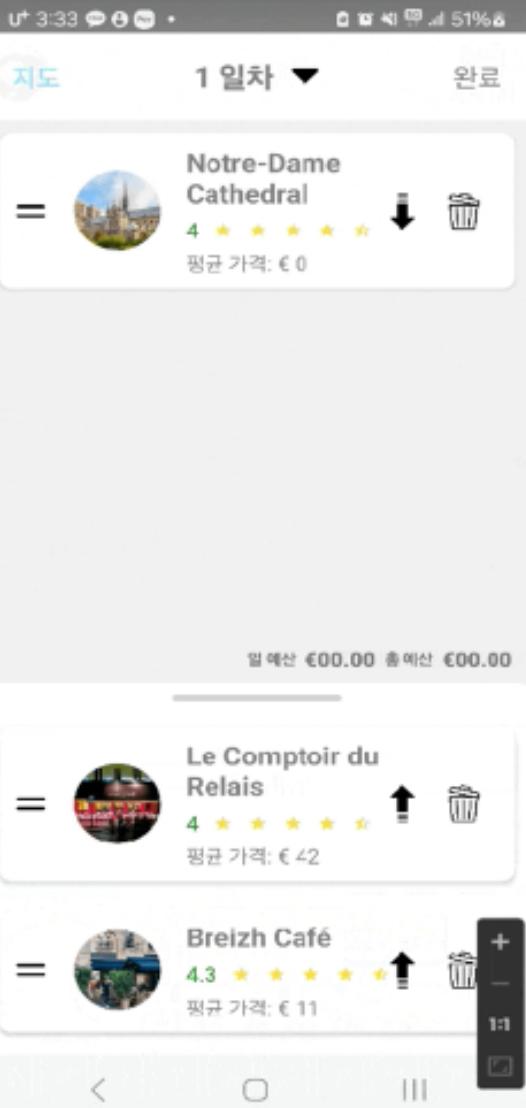
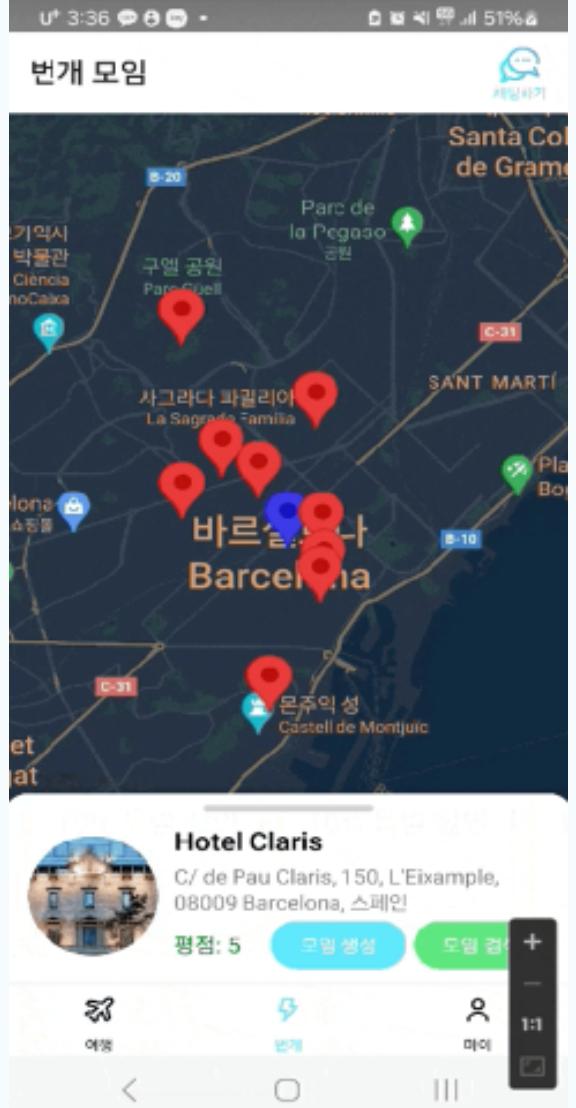
Solution : 전략패턴 Strategy Pattern 적용

- 여러 AI는 각각 다른 모양의 데이터를 가짐
 - 자료형 Integer, Float과 이미지 내 (0, 0) 좌표의 위치 등
 - **각각의 데이터**를 처리해주는 **다른 전략의 로직** 필요
- **런타임에 로직을 정하는 전략 패턴** 적용
 - MoveNet, Openpose의 결과를 처리해주는 클래스 각각 생성
 - 처리 클래스들을 Map으로 저장
 - AI 결과 데이터의 key로 사용할 클래스를 지정

Result

- 여러 종류의 AI 결과 데이터를 처리하는 효율적인 코드로, **가독성 향상**
- 새로운 AI에 대한 **확장성** 확보
- 각각의 AI 데이터 **테스트에 용이**

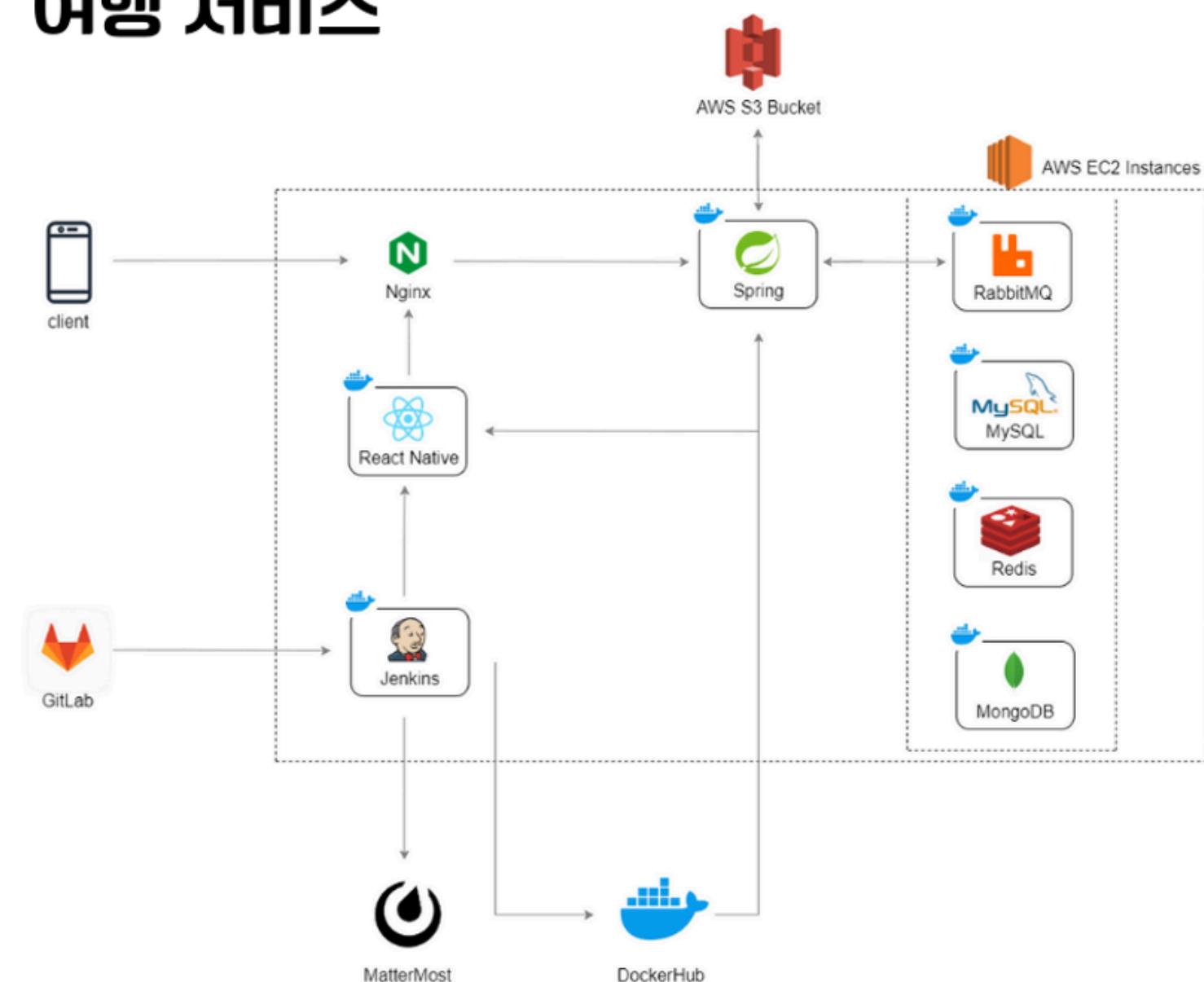
Trip-Together

기간	2024.02.26 ~ 2024.04.05				
팀 구성	Back-end 3인 Front-end 3인				
기여도	Back-end 40% Infra 50%				
핵심 기능	<ul style="list-style-type: none">• 가상은행을 구현. 은행과 trip-together 계정을 연동하여 사용.• 화폐 별 여행 지갑 생성 기능• 장소, 축제 기반 번개모임 채팅방 기능• 은행 거래 목록 기반 정산 기능• 장소, 축제 기반 여행계획				
기술	서버 Java, Spring 인프라 Jenkins, GitLab, Docker				
소스코드	Trip-Together				

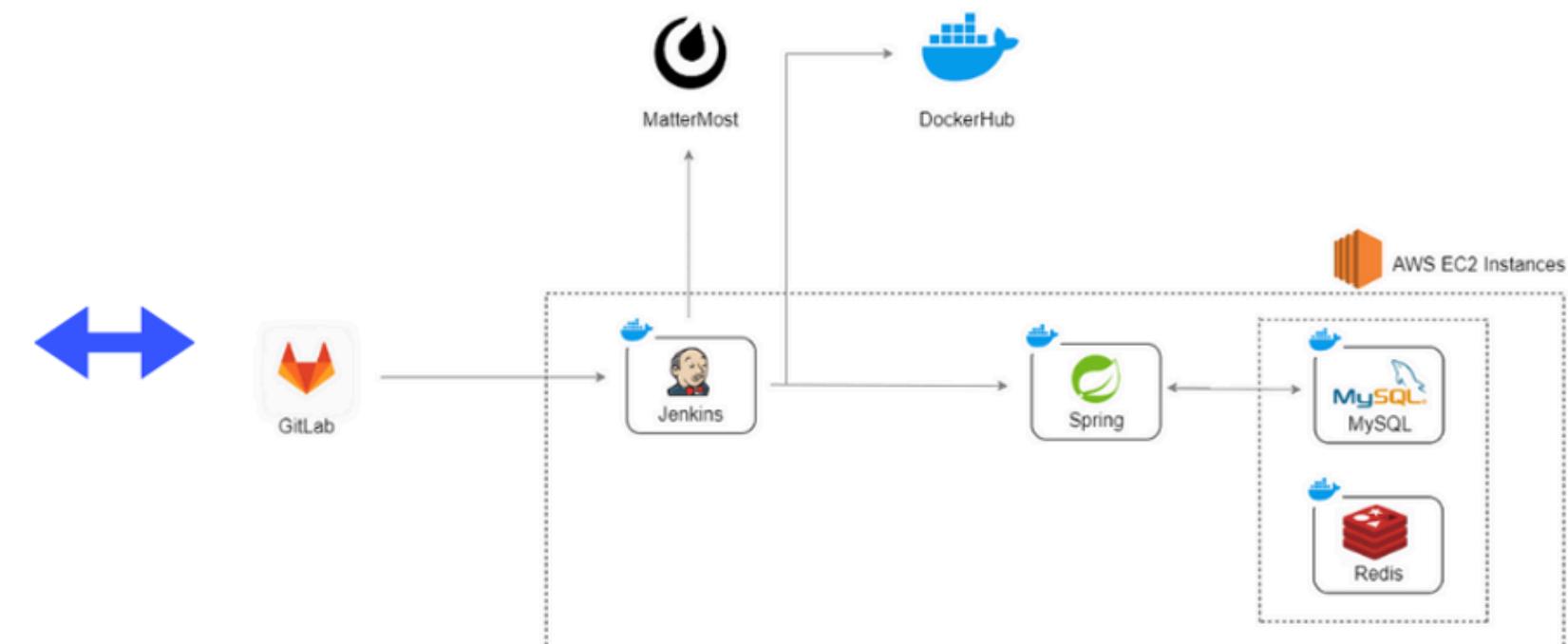
Trip-Together

System Architecture

여행 서비스



은행 서비스

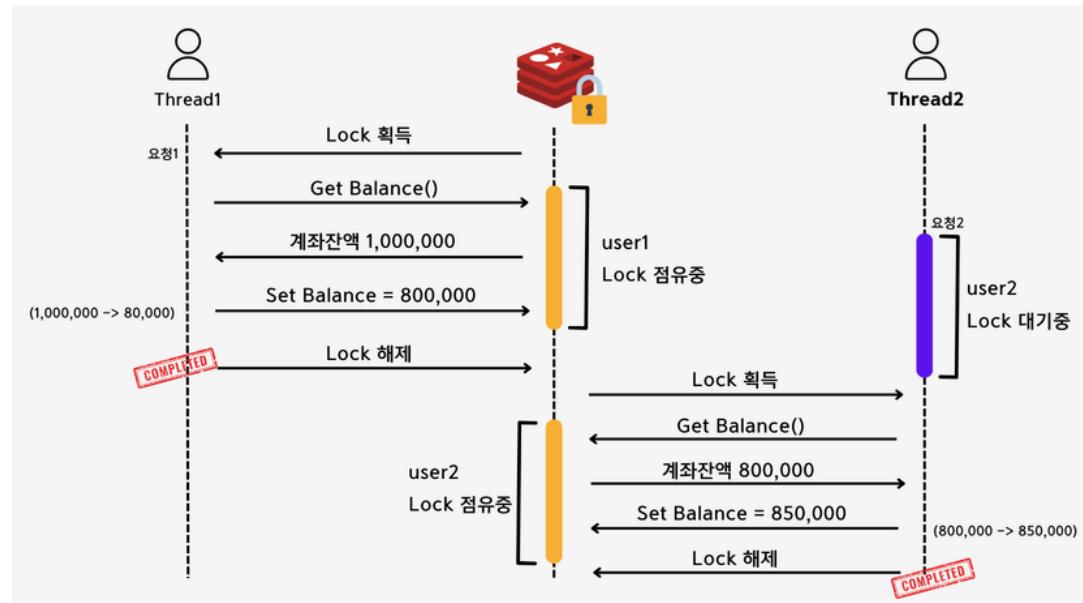


Trip-Together

Issue : 동시성 처리

Problem : 여러 스레드에서 동시에 DB에 접속

- 스프링 서버의 특성상 여러 스레드를 스레드 풀로 만들어 서버 실행
- 가상은행 구현 시, **데이터 정합성**은 중요한 포인트



Solution : Redisson 프레임워크 활용 분산 락 적용

- 동시성 문제를 DB 측의 Pessimistic Lock과, 서버측의 분산 락 중 **분산 락 선택**
 - 모놀로식으로 구현되는 서버가 점점 줄어들고 있는 추세
 - **다중 서버와 분산 데이터베이스에 효과적인** 분산 락 선택
- **Redisson 프레임워크**를 활용하여 구현
 - Redis의 Pub/Sub을 활용한 **효율적인 분산 락**이 적용 가능한 Redisson 적용
 - 다른 프레임워크 활용 시, DB락을 얻기 위한 대기 로직이 스판 락 방식 (비효율)
- **Spring AOP와 커스텀 어노테이션**을 활용하여 구현
 - 로직 코드와 분리하여 구현 가능
 - **유지보수성** 향상

Result

- Database의 **동시성 문제 해결**
- 락이 필요한 메서드에서도 직접적인 **로직 수정 없이 락 추가** 가능
- 락 관련 로직은 따로 모듈화하여 **유지보수성**

Trip-Together

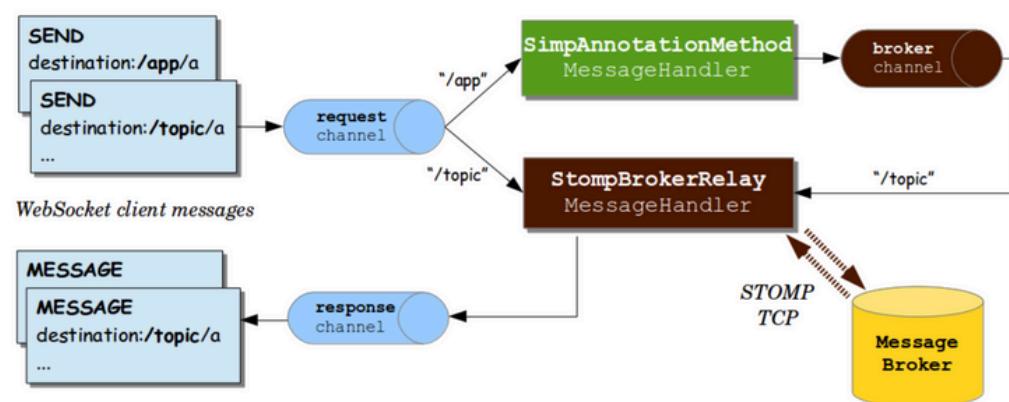
Issue : 실시간 채팅

Problem : 1인 여행을 할 때 여행지에서 많은 경험을 하지 못할 수 있음

- 같은 장소, 축제를 방문할 사람들과 **번개모임** 지원

Solution : 실시간 채팅 기능

- STOMP 프로토콜을 활용해 채팅 구현
 - 메시지 브로커를 활용해 메시지 유실 방지 필요
- RabbitMQ를 메시지 브로커로 선택
 - 해외에서 사용하는 기능인 만큼 **실시간성**을 가장 중요한 요소로 판단
 - 메시지 저장 기간을 지정하여 **효과적인 저장공간** 사용 가능
- 메시지의 종류 세분화
 - 채팅방 입장장, 메시지 전송, 정산 요청 기능을 메시지로 관리
 - 채팅에서 발생한 **모든 이벤트를 메시지 로그로** 간편하게 확인 가능



Result

- 여러 메시지를 **안정적**이고 **빠르게** 보관 및 전송
- Stomp 적용으로 인해 채팅 방과 유저에 대한 유연한 라우팅 가능
- 웹 어플리케이션의 **확장성 확보**

Polaris

기간

2024.01.15 ~ 2024.02.23

팀 구성

Back-end 3인
Front-end 3인

기여도

Back-end 40%
Infra 100%

핵심 기능

- 내 서재에 책 추가, 제거 가능
- 서재를 통해 보유중인 책 확인 가능
- 독서인들을 위한 독후감 게시판
스크랩, 좋아요 가능
- 책 거래를 위한 채팅 기능
SSE를 이용하여 구현
- 배포 서버 중앙화

기술

서버 Java, Spring

소스코드

인프라 Jenkins, GitLab, Docker

POLARIS

이 주의 책
왕좌의 게임 1
조지 R. R. 마틴 (지은이), 이수현 (옮긴이)
전 세계 6천만 독자가 열광한 초대형 베스트셀러 일음과 불의 노래 시리즈가 2016년 1부 왕좌의 게임을 시작으로 개정판을 선보인다. 가상의 대륙 웨스테로스를 둘러싸고 벌어지는 권력과 생존을 위한 투쟁을 다룬 소설이다.

도서 거래를 위한 채팅공간

Chat Room

상대방과 이야기하여 거래도서를 선택해보세요

안녕하세요!

판매도서 목록

판매할 도서를 선택하여 추가해주세요

추가

도서 이름 저자 판매가격

제작주의자 한강 (지은이) 2000원

왕좌의 게임 2 조지 R. R. 마틴 (지은이), 이수현 (옮긴이) 4000원

총 판매금액 : 6000원

판매완료

서점 베스트셀러 Top 20

1위 이처럼 사소한 것들
2위 사랑은 무엇으로 자라는가 - 세계적 가족 심리학자 베르나드 사리어
3위 너를 위한 삼월
4위 나는 메트로폴리탄 미술관의 경비원입니다 - 경이로운 새...
5위 마흔에 읽는 소설 - 마음의 위기와 함께하는 문학

독후감 게시판

해리 포터와 아즈카반의 죄수를 읽고...
월영월영

독극성

무한한 힘을 갖는다는 것은
NULL 사랑하지않아

로버트는

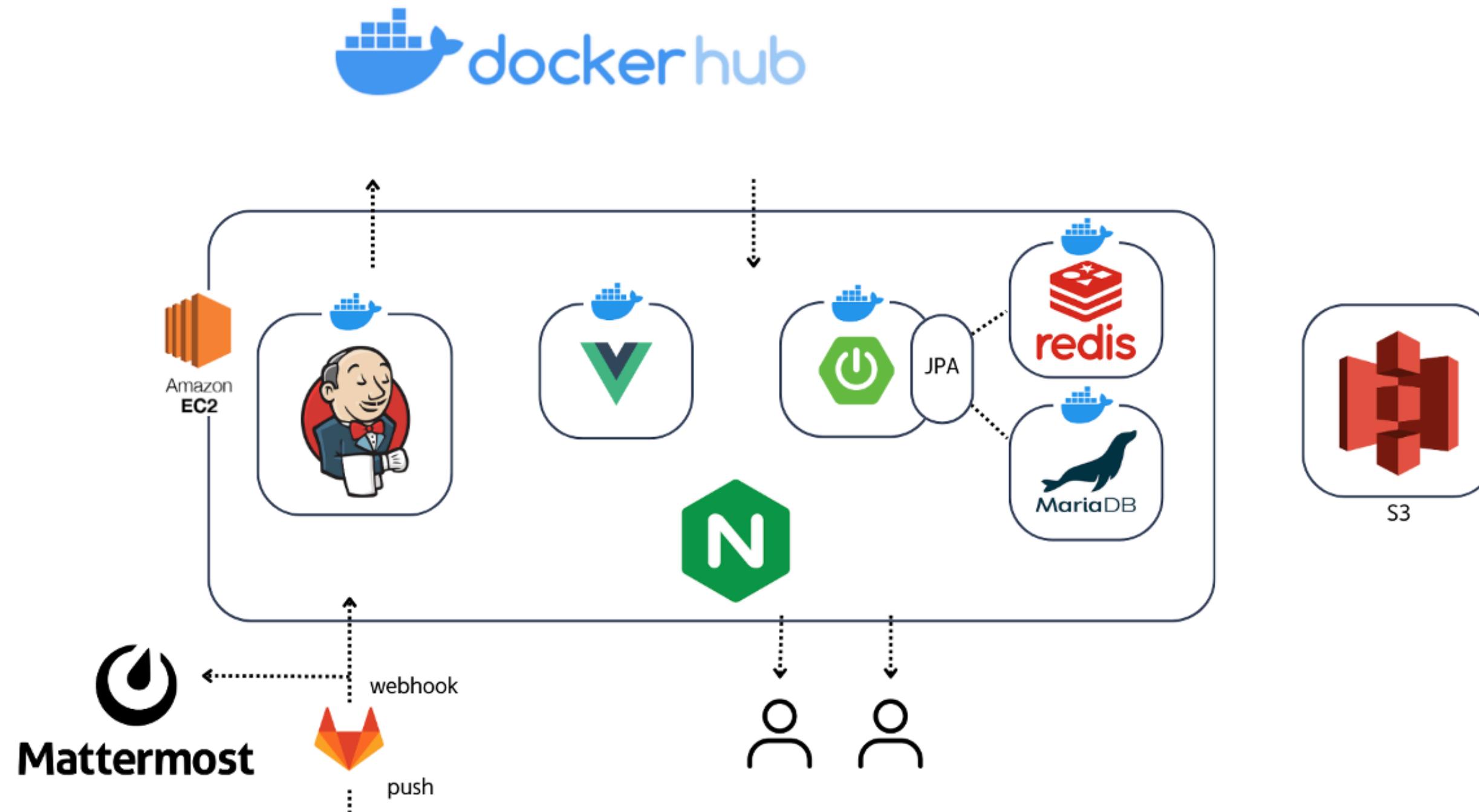
왕좌의 게임 1을 읽고...
명영월월

왕좌의 게임 1
조지 R. R. 마틴

First Previous 1 Next Last

Polaris

System Architecture



Polaris

Issue : 배포 서버 중앙화



Problem : 다중 서버 사용 시 DooD는 한계가 있음

- 매 서버마다 jenkins가 필요하게 되는 문제

Solution : Jenkins에서 SSH 와 Docker Hub를 이용한 원격 배포

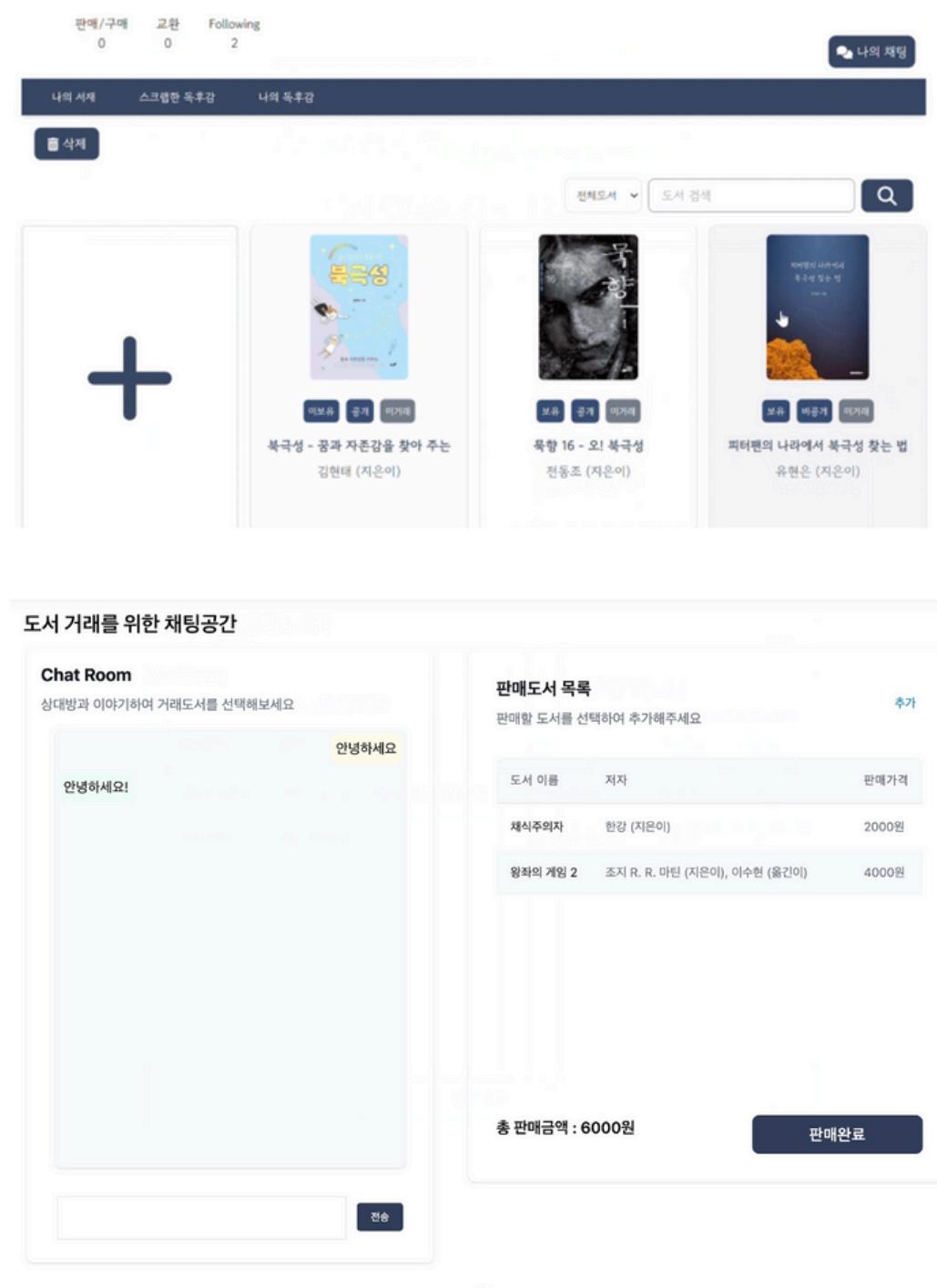
- Jenkins 배포 스크립트 작성
 - 서버 프로젝트 빌드 후 Docker 이미지 생성
 - Docker 이미지 Docker Hub의 내 계정으로 업로드
 - 이미지를 컨테이너화할 서버로 SSH를 활용해 명령어 전송
- credential 관리
 - 배포 서버에 다른 ubuntu서버로 접속하기 위한 RSA키 저장
- Gitlab의 특정 브랜치 push 신호 trigger로 받기
- webhook 사용

Result

- 분산된 서버들의 docker 컨테이너를 중앙서버에서 생성하고 제어하는 방법 학습
- Jenkins를 사용한 배포 자동화 방법 이해
- 각 서버에 jenkins 설치 및 설정하는 과정 간소화 가능. 확장성 향상

Polaris

Issue : 편리한 책 교환/판매 로직



Problem : 책 교환 / 판매 로직을 가장 간편하게

- 내가 관심있는 책을 가진 다른 사람들이 존재하는지 쉽게 볼 수 있어야 함
- 의견교환 및 책을 교환할 쉬운 인터페이스 필요

Solution : 내 서재 & 실시간 채팅 기능

- **내 서재 기능** : 나와 다른 사람들의 보유 책을 쉽게 볼 수 있는 기능
 - 이를 통해 교환/구매 하고자 하는 책을 가진 사람을 찾을 수 있음
 - 프로필이 SNS의 성격을 띠도록 팔로우 기능을 추가
 - 보유한 책에 대해 **독후감** 작성 가능. 서비스 내 모든 독후감을 **게시판과 연동**
- **실시간 채팅 기능** : 책 교환 및 의사소통을 위한 기능. **SSE**를 통해 구현
 - 판매 구매자들 간 의견 교류 가능
 - 교환/판매할 책을 쉽게 선정 가능한 **인터페이스**를 추가한 채팅방

Result

- 단순한 책 중고상점이 아닌, **독서인들을 위한 SNS 및 커뮤니티**로 플랫폼 개발
- **모든 독후감들을 게시판에서 모아볼 수 있으며, 서재에서 도서별로 볼 수 있음.**
- 채팅과 통합되어 간단하고 쉬운 책 교환 인터페이스

End of Portfolio

감사합니다.