

2022  
|  
2025

WORKTYPE

Back-End Developer  
DevOps

# PORTFOLIO



성장을 멈추지 않는 문제 해결형 개발자 오영수 입니다.

CONTACT

010 5178 4922

paul9119298@gmail.com

# + CONTENTS

## 자기소개

- 01 Who am I?
- 02 HISTORY

## 프로젝트

- 01 Hi-Buddy
- 02 Dr.Rate



# 자기 소개



01 who am I ?

02 연혁



## 지속적인 성장과 통합적 역량을 갖춘 개발자 오영수 입니다.

**B** 1998.04.23 (27세)  
**P** 010 5178 4922  
**M** paul9119298@gmail.com  
**G** <https://github.com/YoungSuOh>

### EDUCATION

2025.08 인하대학교 졸업 예정  
2019.08 학점은행제 전문학사 수료  
2018.06 청주대학교 중퇴  
2017.02 분당중앙고 졸업

### CERTIFICATION

2025.01 NCA  
2024.12 NCP  
2024.04 SQLD  
2024.04 OPIC (IM3)

### PROGRAM SKILL

- Java / Spring Boot  
Back-End 개발의 주 언어와 프레임 워크, 프로젝트 경험 3회  
MVC 패턴 기반 CRUD 구현  
JPA와 QueryDSL을 활용한 동적 쿼리 메서드 구현  
Spring Security와 JWT를 활용한 인증/인가 로직 개발  
Websocket과 Apache Kafka를 이용한 채팅 기능 구현  
Naver Clova 및 소셜 로그인을 포함한 외부 API 활용 경험 보유
- Database  
MySQL를 사용하여 join, fetch와 index를 활용한 데이터 최적화 경험  
NoSQL인 MongoDB와 Redis를 사용하여 Document 기반 설계를 통한  
효율적인 데이터 조회와 성능 최적화 진행
- CI/CD  
AWS EC2, RDS, S3를 활용하여 프로젝트 배포 경험  
Naver Cloud Platform와 Jenkins를 활용하여 배포 자동화 및 Docker  
& Dockerhub 를 통한 효율적인 배포 자동화 파이프라인 구축

# HISTORY

## POINT

- ☐ 지속적인 성장
- ☐ 개발에 대한 통합적 접근

## 2022

대학교 복학 후 학업에 성실히 임함과 동시에 동아리를 가입해 스터디를 진행  
방학 기간에는 계절학기, 알고리즘과 실무 기술에 대한 학습을 심화하여 역량 강화



## 2023

실무 기술 학습을 바탕으로 학기 중에는 관련 스터디를 진행 후, 방학 기간에 Android와 Spring Boot 프로젝트를 통해 공모전에 참여하며 협업과 실무 역량을 강화



## 2024

인하대학교 캡스톤 디자인 프로젝트에 참여해 협업 경험을 바탕으로 Spring Boot를 활용한 Back-End 개발을 주도적으로 수행하며 성공적으로 프로젝트를 완수



## 2025

Back-End 개발의 틀에서 한층 성장하기 위해 네이버 클라우드 캠프 과정을 통해 여러 레거시 프로젝트와 최종 프로젝트를 진행하며 Front-End와 CI-CD 역량 강화



## 지속적인 성장과 통합적 역량을 갖춘 개발자

Back-End 개발을 기반으로 하되, 다른 영역에 대한 이해 없이 온전한 개발자가 될 수 없다는 점을 깨달았습니다.

이에 네이버 클라우드 캠프 과정을 통해 Front-End와 CI/CD 등 다양한 기술 스택을 학습하며 개발 전반에 대한 폭넓은 시야를 키웠습니다.

여러 레거시 프로젝트와 최종 프로젝트를 수행하며 팀워크와 문제 해결 능력을 쌓았고, 이를 통해 지속적으로 성장하며 통합적인 개발 역량을 갖춘 개발자로 나아가고 있습니다.

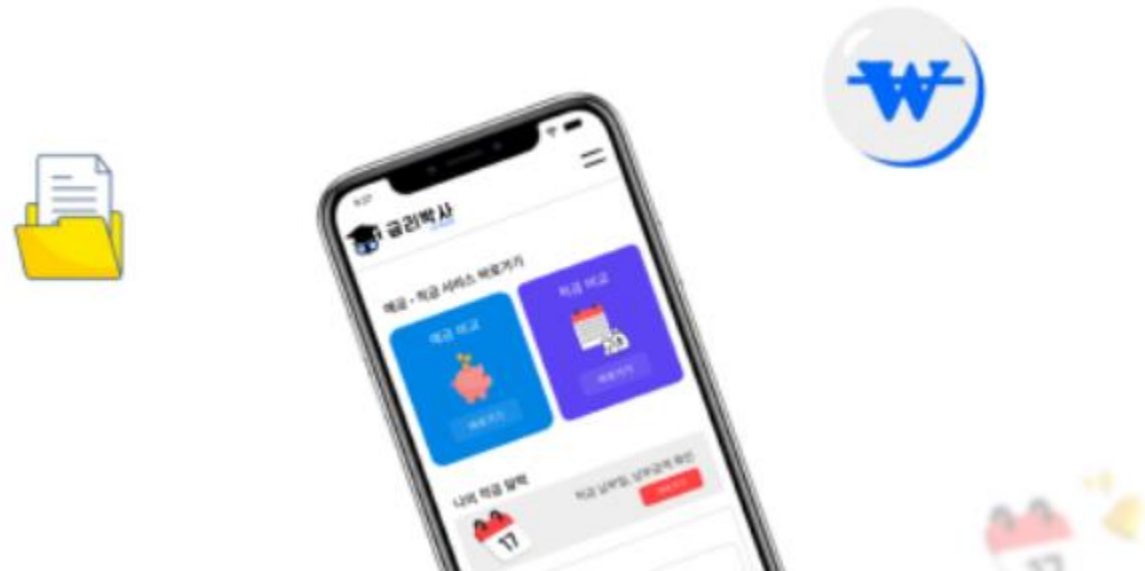
# 프로젝트



01 Dr.Rate

02 Hi Buddy

# Dr.Rate



## 금융 상품 비교 플랫폼

네이버 클라우드 캠프 최종 프로젝트

- 다양한 은행의 예금 및 적금 상품을 검색, 비교할 수 있는 금융 상품 플랫폼
- 사용자 서비스와 관리자 서비스를 통해 금융 상품의 등록, 관리 및 사용자 맞춤형 서비스를 제공합니다.

### TEAM

- Full-Stack Developer 8명, CI-CD 1명
- 역할 : 팀장, Full-Stack Develop & CI-CD

### PROGRAM SKILL

**F** <https://github.com/YoungSuOh/Dr.Rate-Frontend.git>

**B** <https://github.com/YoungSuOh/Dr.Rate-Backend.git>



# MY PART - FRONT END DEVELOP

01

## 관리자 페이지

- 대시 보드 페이지
- 사용자 정보 조회 페이지
- 1:1 문의 내역 조회 페이지 & 1:1 문의 상담 페이지
- 관리자 페이지 헤더를 포함한 모든 페이지 모바일 반응형을 고려한 UI 설계

## 관리자 페이지 화면

대시보드



사용자 정보 조회

1:1 문의 내역 조회

1:1 문의 상담 페이지



# MY PART - BACK END DEVELOP

---

01

## 관리자 대시보드 방문자 현황 집계

- Front-End 측에서는 웹 브라우저의 local storage를 활용하여 방문자 현황 집계
- Backend 측에서는 매일 Redis에 방문자 현황을 TTL을 설정하여 저장하고, 자정이 되면 Spring @Scheduler를 통해 매일 방문자 현황을 저장

02

## 예금 & 적금 필터링을 통한 상품 조회 기능

- 은행, 기간, 이자 계산, 가입 방식 등에 따른 필터링
- QueryDSL을 이용하여 동적 쿼리 처리를 하여 필터를 선택함에 따라 쿼리문을 동적으로 변환

03

## 관리자 1:1 채팅 상담 기능 구현

- Websocket을 이용한 실시간 통신 구축
- STOMP protocol을 사용하여 publish/subscribe 처리
- Message Brocker 용으로 Apache Kafka를 사용하여 Topic을 생성
- MongoDB를 사용하여 채팅방, 채팅 내역, 채팅 이미지 저장

# MY PART – CI/CD

---

## 01

### 클라우드 구성

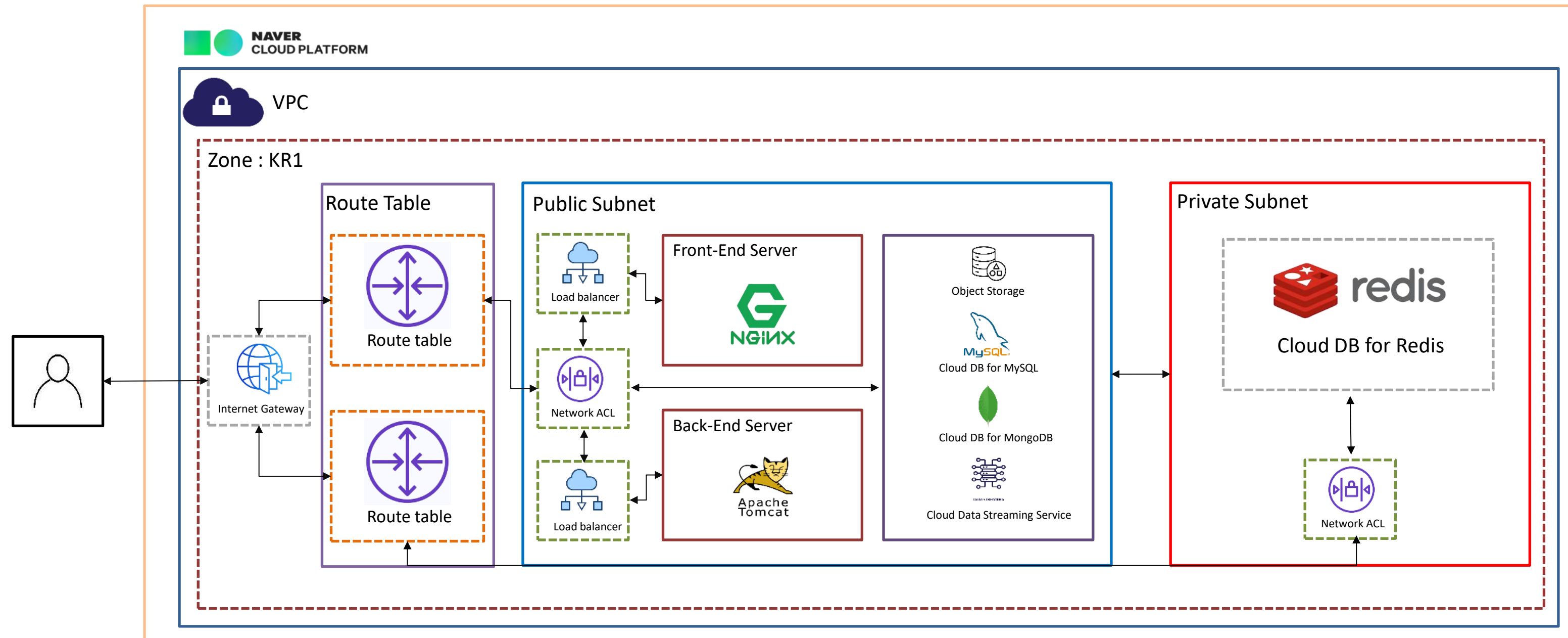
- Naver Cloud Platform를 기반으로 VPC를 설계하여 구성
- Front-End Server는 nginx, Back-End Server는 Spring Boot 내장 tomcat을 이용하여 서로를 따로 운용
- 각 Server를 target group으로 설정하여 로드 밸런서를 연결 후, 도메인을 발급받고 Certification Manager를 통해 SSL 인증서를 등록하여 HTTPS 통신이 가능하도록 함
- 이후 Server Image와 InitScript를 설정 후 AutoScaling을 진행하여 트래픽 분산 처리 진행

## 02

### Jenkins를 통한 배포 자동화

- Jenkins Pipeline을 통해 Github Credential를 등록하여 지속적인 통합 및 배포를 진행
- Docker & Docker hub를 통해 Docker의 재사용성을 활용하여 배포 과정을 효율적으로 자동화
  - Pipeline 과정에서 Docker Image를 Dockerhub에 push를 하여 지속적으로 버전 관리를 진행하고, Docker의 재사용성 활용

# SYSTEM ARCHITECTURE



# LEARNED FROM PROJECT

---

## 인프라 구성

01

- Docker와 Jenkins를 활용한 CI/CD 파이프라인 구축
  - Docker 사용을 통해 이미지와 컨테이너 개념을 이해하고, 애플리케이션 배포 환경을 컨테이너로 통합하는 흐름을 깨달았습니다.
  - Jenkins를 사용하면서 Pipeline을 구축하여 빌드, 테스트 배포를 자동화하는 방법과 Credential 등록, 환경 변수(application.yml) 관리 방법을 학습하였습니다.
- LoadBalancer와 AutoScaling을 진행하면서 확장 가능한 인프라의 중요성에 대해서 깨닫고, 서버 확장할 때 고려해야 될 여러 요소들에 대해서 경험하게 되었습니다.

## 브라우저 Storage 활용

02

- 프로젝트를 진행하면서 기존의 Back-End 개발 경험 외에도 브라우저 Storage인 Local Storage를 활용한 경험을 얻었습니다. 이를 통해 방문자 현황을 집계하는 기능을 구현할 수 있었고, 브라우저 Storage를 실무에서 활용하는 방법을 배웠습니다.
- 또한, JWT Token 정책에 대해 논의하며 Cookie와 Session Storage에 대해서도 학습하고, 각각의 사용 사례와 보안 측면에 대해 깊이 이해하게 되었습니다.

# LEARNED FROM PROJECT

---

## 실시간 채팅 기능 구현

03

- 실시간 채팅 기능을 구현하면서 WebSocket을 활용한 양방향 통신의 원리를 이해하게 되었습니다.
  - 기존의 HTTP 통신 방식에서 벗어나 실시간으로 데이터가 주고받는 방식에 대해 알게 되었습니다.
- Apache Kafka를 활용해 Topic을 생성하고 삭제하는 방법을 익히며, 채팅 기능에서 발생한 Metadata 갱신 문제를 Kafka의 환경 설정과 HTTP Polling 방식을 결합하여 해결했습니다.
  - 이 과정에서 발생한 여러 어려움을 해결하며 실시간 통신 개발에 대한 깊은 이해를 얻을 수 있었습니다.

# Hi Buddy

## 국제학생 커뮤니티 서비스

인하대학교 캡스톤 디자인 프로젝트

- 인하대학교 국제학생들의 정보 격차를 해소하고 한국 생활 적응을 돕는 웹 커뮤니티 플랫폼
- 교내 생활 필수 정보 제공과 함께 국제학생 간 소통을 지원하며, 한국어 발음 평가 기능을 통해 언어 적응을 효과적으로 도와줍니다.

### TEAM

- Front-End Developer 1명, Back-End Developer 2명
- 역할 : Back-End Develop & Deploy

### PROGRAM SKILL



G <https://github.com/YoungSuOh/HiBuddy-BE.git>

# MY PART - BACK END DEVELOP

---

01

## 회원 로그인

- OAuth2.0 소셜 로그인 (카카오, 구글)
- JWT 인증을 통한 Stateless 서버 아키텍처 구축
- Spring Security를 통한 인증/인가 로직 설계

02

## 마이페이지

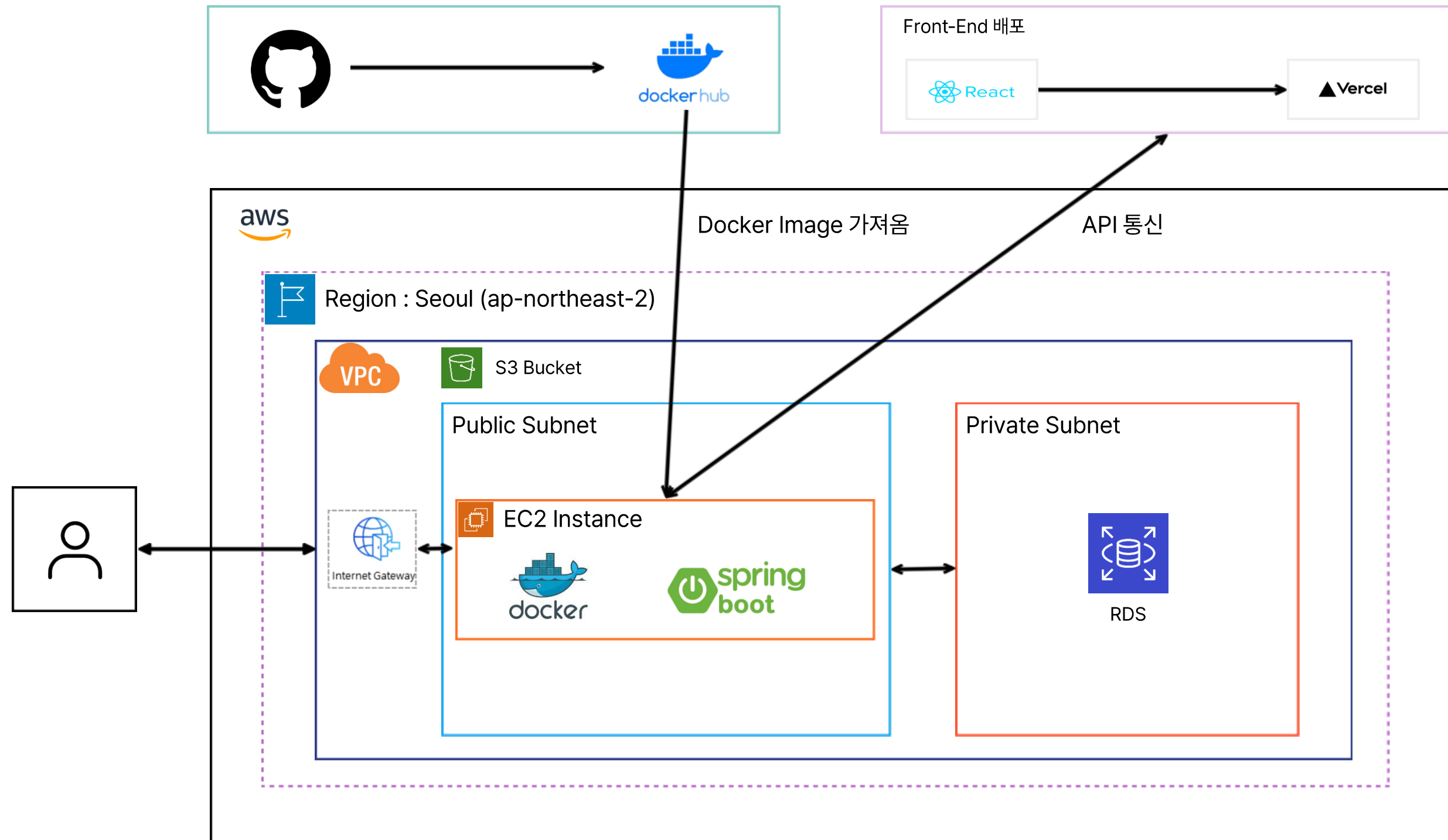
- 프로필 수정 (닉네임, 프로필 이미지)
- 작성한 게시글 및 스크랩 게시글 조회
- 로그아웃 기능

03

## 한국어 발음 평가 & 발음 평가 히스토리 조회 기능

- 한국어 발음 평가 점수 산출
  - Naver Clova API를 음성을 텍스트로 변환
  - Levenshtein Distance 알고리즘을 이용하여 스크립트와 변환된 텍스트 사이의 각 문자열 간 편집 거리 계산을 통한 점수 산출
- 난이도별 스크립트 평가 기능
  - 난이도는 상, 중, 하 스크립트 선택 가능
  - 국제언어문학회에서 발표한 논문에 의거하여 외국인이 발음하기 어려운 자음, 모음을 토대로 설정
- 한국어 발음 평가 내역 조회 기능

# MY PART - BACKEND DEPLOY



01

## EC2 Instance

- 외부와 통신하기 위해 EC2 Instance를 public subnet에 배치
- 이후 Github에 Dockerfile을 생성해 image를 Dockerhub에 업로드 한 뒤, EC2 Instance에서 해당 Image를 가져와 컨테이너화 진행

02

## MySQL RDS

- MySQL RDS를 사용하여 데이터베이스 구축



# LEARNED FROM PROJECT

---

## 새로운 기술과 개념에 대한 도전

01

- 기존 HTTP 클라이언트인 RestTemplate, WebClient를 사용하는 것이 아닌 Spring Boot 3.2부터 신규로 도입된 HTTP Client인 Restclient 를 공식문서를 참조하여 사용해보았습니다.
  - 이는 블로그 또는 chatgpt와 같은 곳에서 정보를 얻을 수 없기에 스스로 최신 기술을 도입해보는 값진 경험이 되었습니다.
- OAuth2.0 소셜 로그인 인증/인가 로직을 설계하는 과정에서 각 API의 인증 흐름을 익히고 Spring Security 설정의 중요성을 깊이 이해할 수 있었습니다.
- JWT 토큰의 stateless 아키텍처 개념을 이해하고 구현하는 데 어려움이 있었지만, Access Token과 Refresh Token의 역할을 명확히 알게 되었습니다.

## 팀원과의 협업 및 문제 해결

02

- 같은 백엔드 팀원의 진행 속도가 더뎠던 프로젝트 마감 일정 기한을 못 맞추출 뻔했지만, 적극적인 소통을 하여 현재 진행과정과 어려움에 대해서 이야기 했고, 이를 통해 진행 과정을 이어받아 서로 역할 분담을 재정비하여 프로젝트를 성공적으로 완수했습니다.
  - 이를 통해 프로젝트에 대한 책임감과 문제 해결 능력을 크게 키울 수 있었습니다.

# LEARNED FROM PROJECT

---

## 외부 API와 알고리즘 적용

- 처음으로 외부 API를 사용하면서 access key 사용법을 익히고, 공식 문서를 통해 API의 요청 및 응답 구조를 분석하여 어떻게 프로젝트에 적용해야 되는지 알게 되었습니다.
- Levenshtein Distance 알고리즘을 적용하기 위해 학습을 진행하면서, 2차원 DP 테이블을 활용한 최소 편집 거리 계산 방식을 익혔고, 이를 통해 다양한 NLP 분야에서 활용 가능한 알고리즘적 사고를 기를 수 있었습니다.
  - 또한, 난이도별 스크립트를 설계하는 과정에서 연구 자료를 기반으로 객관적인 지표를 적용하여 개발의 신뢰도를 높이는 방법의 중요성을 깨달았다.
  - 이를 통해 단순한 기능 구현을 넘어 데이터 기반의 의사 결정과 평가 시스템 설계가 AI 서비스에서 핵심적인 요소임을 인식할 수 있었습니다.

# THANK YOU



끝까지 봐주셔서 감사합니다.  
꿈임없이 성장하는 개발자 오영수의 포트폴리오였습니다.