

AWS Cloud School 과정

## 프로젝트 완료 보고서

2024.03.31

개인별 운동 경기장 대여 웹 앱 개발  
TukTuk

### 프로젝트 완료 보고서

프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발

단계 : 프로젝트 완료

작성자 : XXX

작성일 : 2024. 03. 31

개정이력

개정번호	개정일자	시행일자	개정내용	담당자
1.0	24.03.22	24.03.22	최초 작성	XXX
1.1	24.03.23	24.03.23	1차 작성	XXX
1.2	24.03.25	24.03.25	2차 작성	XXX
1.3	24.03.27	24.03.27	3차 작성	XXX
1.4	24.03.31	24.03.31	작성 완료	XXX

주관기관	한국전파진흥협회
주관사업자	TukTuk
용역책임자	프로젝트 총괄 XXX
실무참여자	프로젝트 관리 XXX
	웹 개발 XXX
	인프라 구현 XXX
	웹 디자인 XXX

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 목 차

1. 프로젝트 소개
2. 팀 소개
3. 문제 및 요구 사항 정의
4. 주요 기능
5. 프로젝트 구성
6. 개발 과정
7. 협업 방식
8. 회고

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 1. 프로젝트 소개

본 프로젝트는 운동 경기를 하고 싶은 개인들을 매칭해주거나, 동호회와 같은 단체를 위한 경기장 대여도 가능한 플랫폼 서비스입니다.

서비스의 전체적인 구성은 3 Tier Architecture 형태로, 사용자와 상호작용하는 웹 계층은 React를 사용하여 구현하였으며 비즈니스 로직을 처리하는 웹 어플리케이션 계층은 Spring을 사용하여 구현하고, MySQL에 데이터를 저장하도록 구성하였습니다.

서비스가 동작하는 인프라 환경은 AWS EC2 인스턴스를 사용하여 웹 어플리케이션을 배포하였습니다. 또, AWS EKS 클러스터를 구성하고 ELB와 Auto Scaling 그룹을 사용하여 웹 서버의 확장성 및 안정성을 보강했습니다.

또한, Github Action과 Argo를 이용하여 CI/CD 파이프라인을 구축하여 개발 과정에서 작성되는 코드를 빌드하여 배포 환경에 배포되도록 자동화하였으며, Prometheus와 Grafana를 사용하여 모니터링까지 가능한 환경을 마련하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 2. 팀 소개

### 프론트엔드

- 사이트/앱 최적화를 위한 비동기 통신 구현
- API를 통해 서버와 데이터 통신 구현
- CORS 에러 핸들링
- 웹 페이지 제작
- 로고 및 배너 제작
- 입력 폼 디자인

### 백엔드

- 도메인 및 데이터베이스 스키마 설계
- 개발 편의를 위한 스크립트 및 아티팩트 생성을 위한 Dockerfile 작성
- 경기 매칭, 경기장 예약에 대한 API 개발
- 경기장에 대한 API 개발
- 인증 및 인가 기능 구현

### DevOps

- AWS에서 동작하는 K8S 기반 인프라 설계 및 구축
- 코드 기반의 인프라(IaC) 관리
- 카카오 로그인 연동을 위한 Cognito User Pool 설정 관리
- GitOps 기반 CI/CD 파이프라인 구축
- Prometheus를 이용한 메트릭 모니터링
- Loki Distributed 를 이용한 로그 모니터링

## 3. 문제 및 요구 사항 정의

해당 프로젝트에서는 운동 경기장 매치 및 대여 플랫폼 서비스를 구축을 목표로 하였습니다. 이를 위해 사용자가 경기를 매치하는 기능이나 운동 경기장을 등록하는 등의 기능을 수행하기 위한 웹 페이지를 만들어야 하고, 해당 기능을 처리하기 위한 어플리케이션을 구현해야 합니다.

또, 수 많은 트래픽에 대응하기 위해 Auto Scaling이 가능해야 하며 이를 시각화하여 볼 수 있는 모니터링 대시보드가 필요합니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 4. 주요 기능

### AWS EKS + Auto Scaling

클라우드 환경에서 웹 어플리케이션을 자동으로 배포하고 동작시키기 위해 AWS EKS 기반의 클러스터를 구성하였습니다. 구체적으로는 EKS의 Auto Scaler를 EC2의 Scaling Group으로 활용하여 주기적으로 파드의 상태를 확인하고 지정된 상태로 변화하는 방식으로 동작하도록 구성하였습니다.

### AWS ECR

도커 이미지를 관리하는 수고를 덜기 위해 완전 관리형으로 제공되는 AWS ECR를 사용하여 React application, Spring application, MySQL을 각각 하나의 도커 이미지로 저장하여 관리하였습니다.

### Ingress + ALB

Kubernetes ingress와 ALB를 연계하여 웹 어플리케이션 트래픽을 적절히 로드 밸런싱이 되도록 구성하였습니다. 이번 프로젝트에서는 ingress를 Auto Scaling 그룹으로 설정하여 웹 서버의 고가용성을 보장하였습니다.

### RDS + InnoDB Cluster

본 프로젝트는 운영 환경 및 개발 환경을 구축하였습니다. 개발 환경에서는 InnoDB Cluster를 사용하여 데이터베이스를 배치하였고, 운영 환경에서는 AWS RDS를 사용하여 다중 가용 영역의 데이터베이스를 배치하였습니다. 비용 효율적인 측면에서 개발 환경에서는 InnoDB Cluster만을 사용하였고, 실제 운영 환경에서만 AWS RDS를 사용하였습니다.

### CI/CD (Github Action + Argo)

Github Action을 사용하여 Git Repository에 코드가 적재되는 이벤트를 기반으로 빌드를 수행하여 아티팩트를 생성하고, 해당 아티팩트를 실행하는 도커 이미지를 빌드하여 ECR에 저장하는 CI 파이프라인을 구축하였습니다. 여기에 Argo CD를 연계하여 배포까지 자동화된 CD 파이프라인을 구축하여 Github에 코드가 적재되면 개발 및 운영 환경에 자동으로 배포되도록 파이프라인을 구축하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

Prometheus, Grafana

모니터링 도구인 Prometheus, Grafana를 사용하여 웹 서비스에서 일어나는 모든 로그를 수집하도록 하고, 이를 시각화하여 볼 수 있는 대시보드까지 구성하였습니다.

Web Application

EKS 위에서 동작하는 어플리케이션을 구현하였습니다. 구체적으로는 사용자와 상호작용하는 react application과 비즈니스 로직을 처리하는 spring application을 구축하여 사용자와의 상호작용을 담당하는 application과 사용자의 요청에 따른 로직 처리를 담당하는 application을 구축하여 서비스를 구현하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

5. 프로젝트 구성

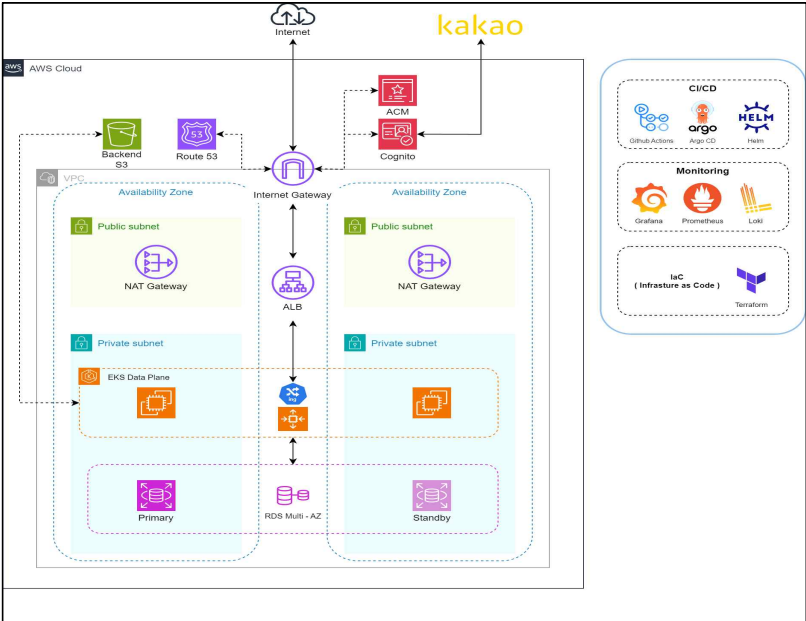


그림 1. 프로젝트 인프라 구성

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

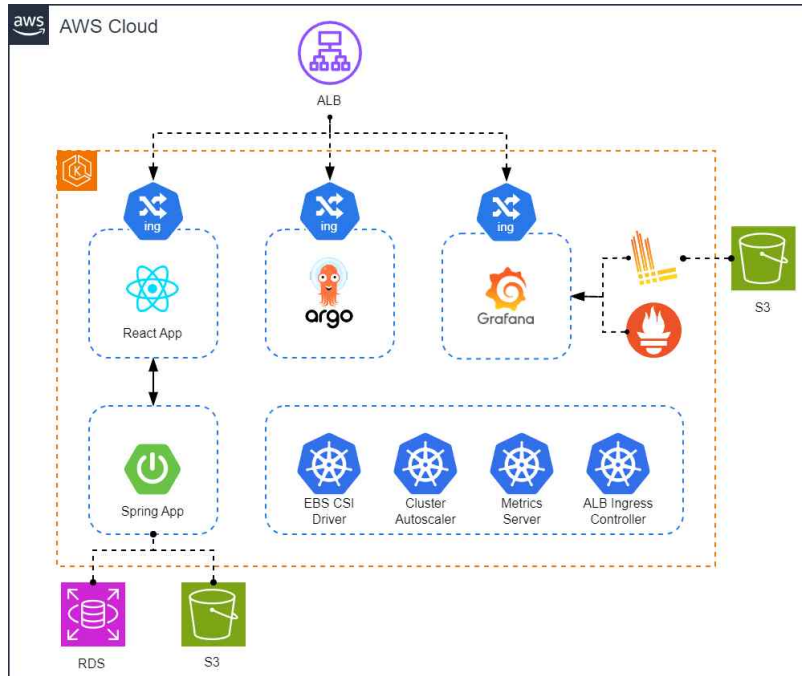
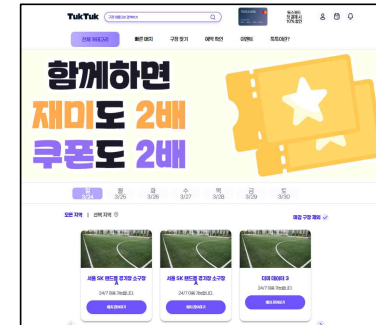


그림 2. EKS 구성

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 6. 개발 과정

### 1. 웹 페이지 디자인 및 페이지 렌더링



POST 경기장 등록 [FIELD_OWNER]			
https://devapi.tuktuk.com/stadiums			
경기장 등록 API입니다. 요청 시에 HTTP Header에 FIELD_OWNER가 포함된 JWT 토큰을 추가하여 요청해야 합니다.			
요청			
Request Body			
요청 변수명	타입	설명	필수 여부
name	String	경기장 이름	Y
province	String	도/시	Y
city	String	시/구	Y
roadNameAddress	String	도로명 주소	Y
specificInfo	String	주차/오른길/안내 등 상세 내용	Y

사용자와 상호작용하기 위한 웹 페이지를 디자인하기 위해 프론트엔드와 백엔드 담당 인원들은 버튼과 같은 컴포넌트들의 배치를 단일 페이지 단위로 고민하고, 페이지마다 호출할 API의 요청과 응답에 대해 논의하며 구현하였습니다.

### 2. 도메인 및 데이터베이스 설계

users			
PK	id	bigint	NOT NULL
이메일(유저 계정)	email	varchar(36)	NULL
닉네임	nick_name	varchar(16)	NOT NULL
성별	gender	boolean	NOT NULL
전화번호	telno	varchar(36)	NOT NULL
생성 날짜	created_at	localDateTime	NOT NULL
시/도	province	Province	NOT NULL
구/군	city	City	NOT NULL
역할	role	Role	NOT NULL
소셜 로그인 기관	provider	Provider	NOT NULL

stadiums			
PK	id	bigint	NOT NULL
이름	name	varchar(36)	NOT NULL
소유주	owner_id	OwnerId	NOT NULL
위치	location	Location	NOT NULL
상세 정보	specific_info	text	NOT NULL

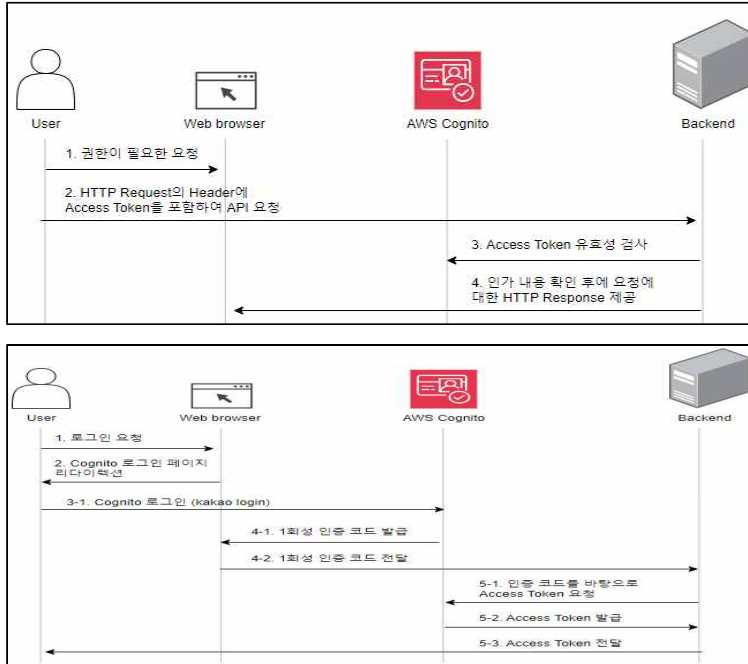
  

courts			
PK	id	bigint	NOT NULL
경기장 ID	stadium_id	StadiumId	NOT NULL
이름	name	varchar(36)	NOT NULL
타입	type	CourtType	NOT NULL
시간당 대여비	hourly_rent_fee	bigint	NOT NULL

백엔드를 담당하는 인원들은 서비스의 요구사항을 구체적으로 도출하면서 애플리케이션에서 사용할 도메인 및 객체, DB에 저장될 데이터 모델을 설계하였습니다. 특히, Spring Data JPA를 사용하였기에 N+1 문제를 방지하고 불필요한 테이블 생성을 줄이는 것에 초점을 두고 설계 작업을 진행하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

### 3. AWS Cognito와 연계한 역할(Role)별 인가 기능 구현

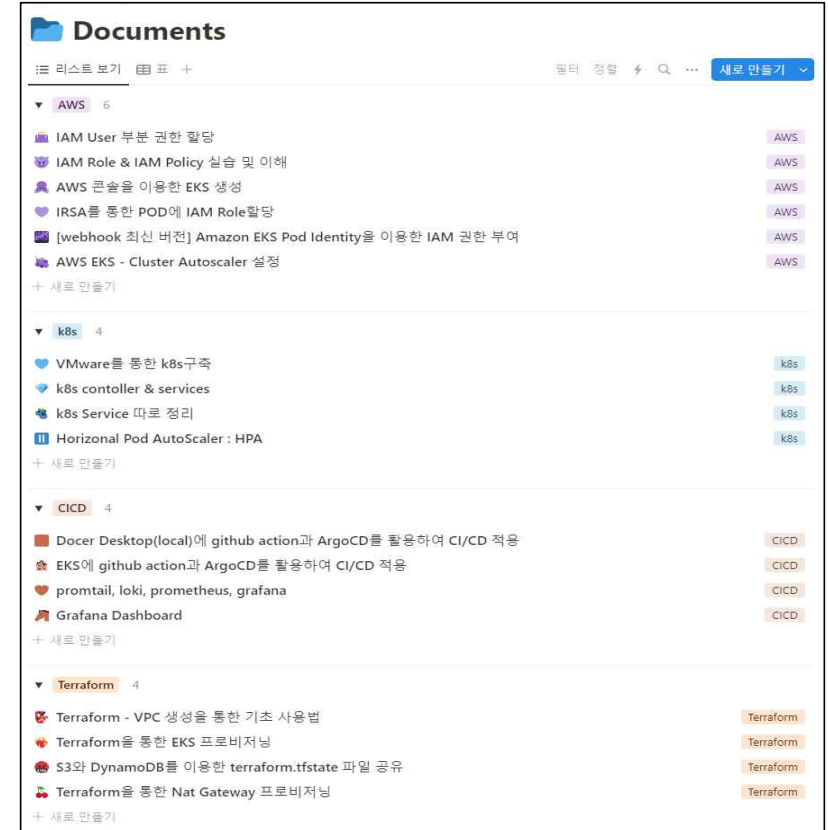


TukTuk은 사용자를 경기장 소유주, 일반 사용자로 구분합니다. 이에 따라 Application으로 오는 수 많은 요청들 중에 요청을 보낸 사용자가 해당 요청을 호출할 권한이 있는 지를 확인할 필요가 있습니다. 이러한 기능을 수행하기 위해서는 인증 및 인가를 담당할 리소스를 개발하고 유지보수 해야하는 번거로움이 있습니다.

이를 줄이기 위해 해당 프로젝트에서는 AWS에서 제공하는 Congnito를 사용하였습니다. Congnito를 통하여 인증의 역할을 대신하도록 하고, Spring Application에서는 Congnito를 통해 만들어진 Access Token을 필터링하는 인가 기능만을 구현함으로써 요청을 보낸 사용자를 식별하여 해당 사용자가 요청을 처리할 수 있는 지를 구현하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

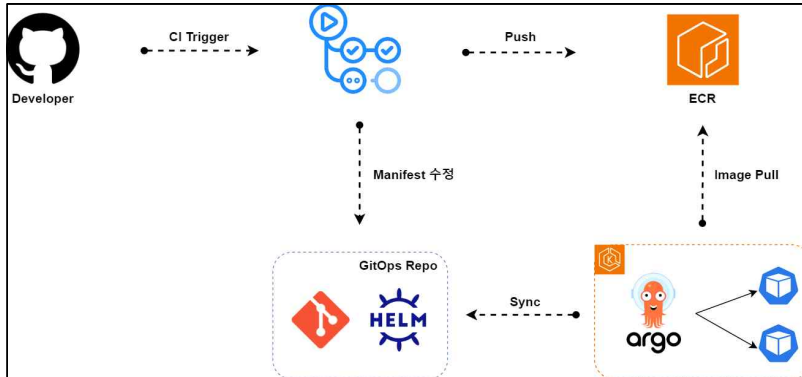
### 4. 운영 및 개발 환경 구축



프론트엔드와 백엔드 팀원들이 Application 개발에 집중하는 동안 DevOps 팀원들은 테라폼을 통하여 AWS 환경에 ALB, Auto Scaling Group 등의 인프라를 개발 환경과 운영 환경으로 나누어 구축하는 작업을 수행하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

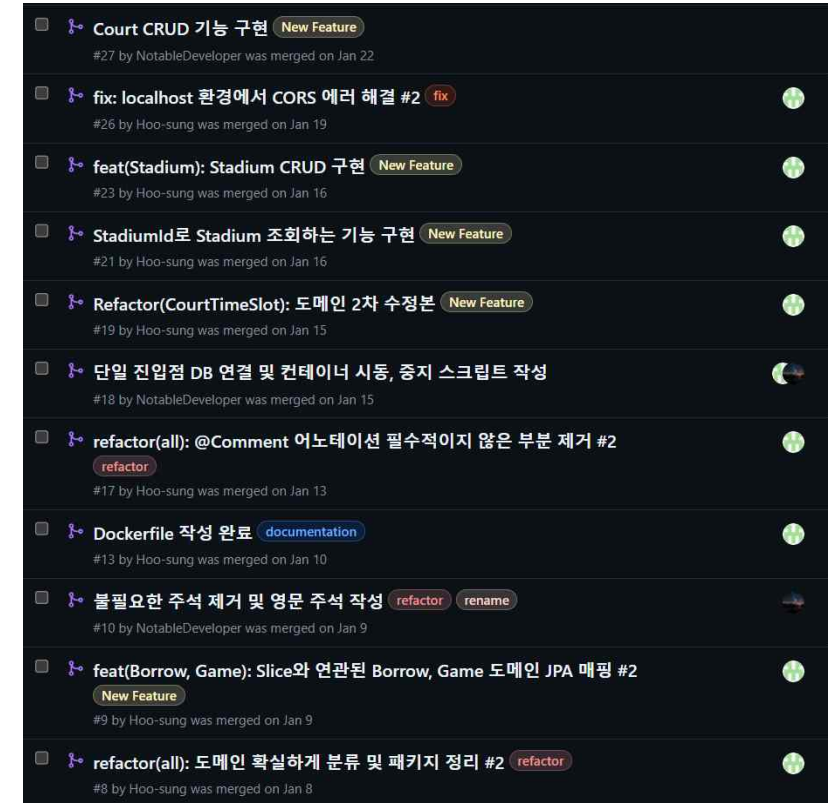
## 5. CI/CD Pipeline 구축



개발 과정에서 아티팩트로 빌드, 도커 이미지로 빌드, ECR 저장, 운영 및 개발 환경으로의 배포를 자동화하기 위해 Git Action과 Argo CD를 사용하여 자동화 파이프라인을 구축하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 7. 협업 방식



모든 팀원들은 Github를 사용하여 Git-flow 방식의 브랜치 관리 전략을 사용하여 기능 개발을 진행하였습니다. 또, 커밋 컨벤션을 적용하여 커밋도 하나의 문서라 생각하여 적절한 커밋 메시지를 남기도록 노력하여 커밋 이력을 추적할 때에 수월하도록 하였습니다.

프로젝트 완료 보고서		
프로젝트명 : 개인별 운동 경기장 대여 웹 앱 개발		
단계 : 프로젝트 완료	작성자 : XXX	작성일 : 2024. 03. 31

## 8. 회고

### 프론트엔드

Argo CD 대시보드를 통해 클러스터의 배포 상태를 시각적으로 확인하고 실시간으로 서버의 오류를 모니터링하여 문제를 쉽게 파악할 수 있었습니다. 이후에는 실제로 결제기능을 포함한 어플리케이션을 개발해보고 싶습니다.

### 백엔드

경기장 검색, 운동 경기 매치 일정 조회와 같은 검색 및 조회 기능을 구현할 때, JPA만으로 구현하는 것에 어려움을 느꼈습니다.

추후 QueryDSL까지 활용하여 다채로운 검색 기능을 구현해보고 싶습니다.

Cognito를 활용하여 인증 서비스를 온전히 AWS에 위임하고, Application에서는 인가 기능만에 집중하여 구현할 수 있어 좋았습니다. 추후에 필요하다면 Cognito와 같이 인가 기능을 위임할 수 있는 서비스나 프레임워크를 공부해보려고 합니다.

### DevOps

DB 운영에서 고가용성과 비용의 트레이드 오프를 경험하며, 비용이 증가하더라도 클라우드 서비스를 활용하여 인프라 아키텍처를 가능한 한 간소화하고 운영 복잡성을 최소화하는 것이 바람직하다고 생각하게 되었습니다. Kubernetes 경험은 처음 배우는 입장에서는 유익하였으나, 향후에는 ECS, Fargate, S3를 포함한 클라우드 서비스를 적극 활용하여 비용 효율성을 극대화하는 방향으로 아키텍처를 재설계해보고 싶습니다.