

안녕하세요,
현장의 불편함을 시스템으로 개선하는 엔지니어
윤상현입니다.

목차

01 프로필 및 이력

- 경력 및 교육 이력
- 대표 프로젝트 흐름

02 기술 스택

- Frontend 구현
- Infra / 배포 환경 구성
- CI/CD 자동화 경험

03 실무 및 문제 해결

- SCADA 모니터링 시스템 운영
- Helium Leak Test 검사 프로세스 개선
- 문제 분석 및 표준화 경험

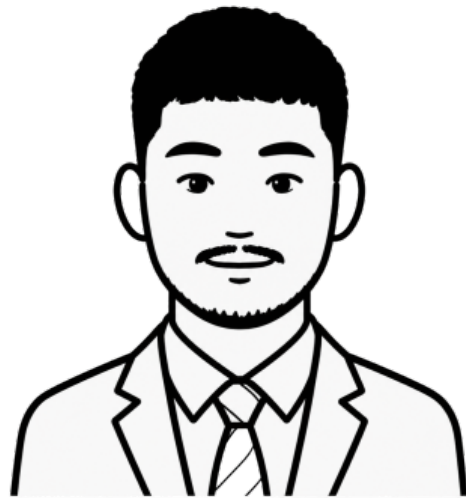
04 주요 프로젝트

- HATER / Egg Studio 웹게임 플랫폼
- MANNEQUIN 멀티플레이 게임
- DDIP 실시간 좌석 예매 서비스

05 성장 방향 및 마무리

- 현장 경험과 개발 역량의 연결
- 프로젝트를 통해 배운 점
- 앞으로의 개발 방향

프로필 및 이력 요약



윤상현 1996.01.19

☎ 010-8563-8335

✉ sangbot96@gmail.com

이력 요약

2025.07~2026.06	삼성청년SW·AI아카데미 14기
2025.03~2025.06	LG U+ WHY NOT SW CAMP 6기
2020.08~2024.11	JH ENGINEERING 재직
2015.03~2021.02	수원대 전자재료공학 졸업

대표 프로젝트

2022.03~2024.11	산업용 SCADA 모니터링 시스템 구축 및 운영 지원
2026.01~2026.02	DDIB: 실시간 좌석 예매 시스템
2026.03~2026.04	MANNEQUIN: PHOTON FUSION 기반 3D 비대칭 PVP 게임
2026.04~2026.05	HATER: UNITY WEBGL 웹게임 플랫폼

커리어 방향성

현장 경험	SCADA/HMI 운영, 장비 모니터링, 원인 분석 경험
개발 경험	REACT 웹 포털, API 연동, DOCKER/NGINX, CI/CD 경험
관심 방향	INFRA / DEVOPS, 서비스 운영 안정화, BACKEND 연동
지향성	현장의 문제를 이해하고, 안정적인 서비스 구조로 연결하는 개발자

현장 경험에서 출발해, 서비스 구현과 운영까지 연결할 수 있는 개발자로 성장하고 있습니다.

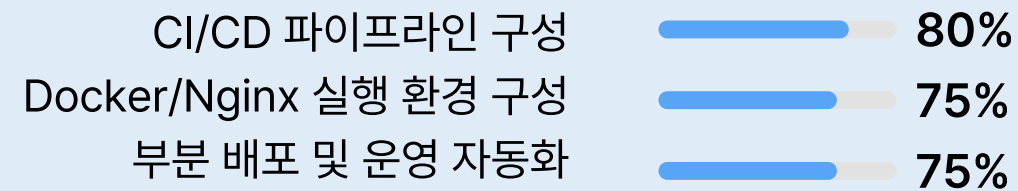
기술 스택 - 구현 및 적용

역할별로 정리한 사용 기술과 프로젝트 내 적용 수준



Infra / DevOps

Docker · Nginx · GitLab CI/CD

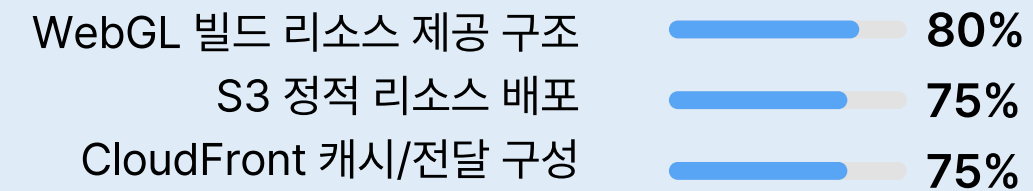


배포 흐름을 분리하고 반복 배포 부담을 줄일 수 있도록 운영 구조를 구성했습니다.



WebGL Resource Delivery

Unity WebGL · S3 · CloudFront

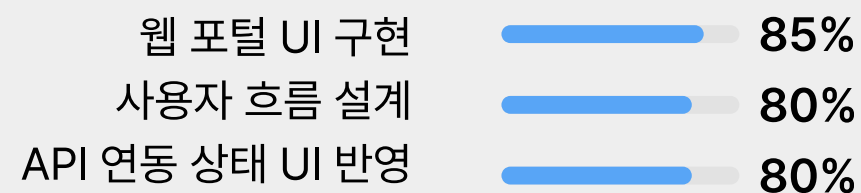


Unity WebGL 빌드 파일을 웹에서 안정적으로 제공할 수 있도록 정적 리소스 구조를 구성했습니다.



Frontend

React · TypeScript · HTML/CSS

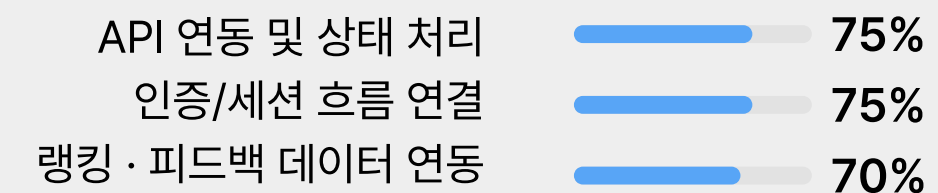


웹 포털 화면에서 게임 소개, 랭킹, 피드백, 마이페이지 흐름을 연결했습니다.



Backend Integration

Spring Boot API · PostgreSQL · Redis

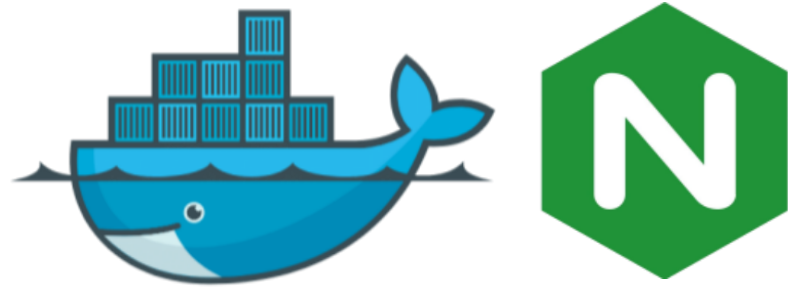


백엔드 응답을 프론트엔드 흐름과 연결해 인증, 세션, 랭킹, 피드백 상태를 화면에 반영했습니다.

기술 스택 · 구현 및 적용

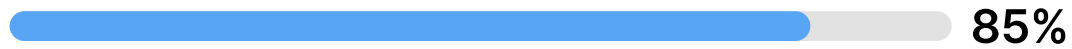
중요 기술 중심으로 정리한 역할별 활용 역량

INFRA / DEVOPS



Docker · Nginx

실행 환경 분리 / 운영 구성

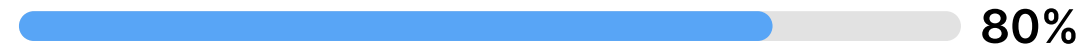


웹 포털과 API 실행 환경을 분리하고
서비스 운영 구조를 구성



GitLab CI/CD

부분 배포 파이프라인 구축

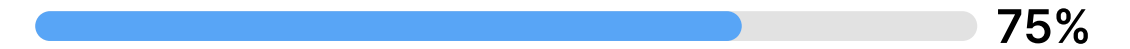


변경 범위 중심 자동 배포 흐름을 구성해
운영 부담 최소화



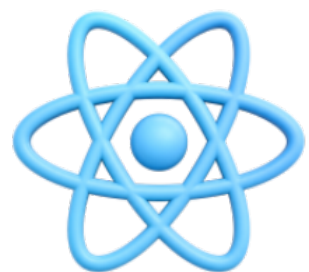
AWS S3 · CloudFront

정적 리소스 배포



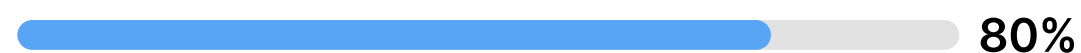
WebGL 빌드 리소스 업로드 및 빠른 정적 리소스 전달

FRONTEND



React

웹 포털 구현

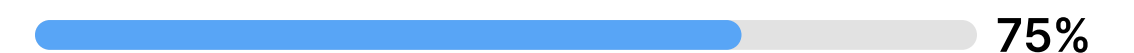


게임 소개, 진입, 랭킹, 피드백 등 포털 화면 구현



TypeScript

UI 구현 / 흐름 설계



컴포넌트 구조화와 사용자 흐름 중심 화면 설계

실무 경험 - SCADA 모니터링

산업용 설비 장비를 실시간으로 모니터링하고,
이상 상황을 빠르게 분석하여 운영자의 판단을 지원했습니다.



역할 및 수행 업무

- SCADA 기반 모니터링 화면 구현
- 알람, 트렌드, 데이터 로깅 효율확인
- 현장/원격 기술 지원 및 문제 대응
- 운영자 피드백 기반 화면 상태 개선

주요 처리 항목

- 설비 상태 실시간 모니터링
- 알람 및 이상 상황 분석
- 트렌드 데이터 확인 및 기록
- 장애 발생시 원인 파악 및 대응



성과 및 개선

- 정보 구조 개선으로 이상 상황 판단 시간 단축
- 알람 기준 및 표시방식 개선으로 오탐 감소
- 로그 및 이력관리 체계 개선

배운 점 및 향후 방향




화면은 단순한 도구가 아니라, 운영자의 판단과 대응을 돕는 정보 구조라는 점을 배웠습니다.
앞으로는 사용자 관점에서 정보의 우선순위와 흐름을 설계하는 개발자가 되고자 합니다.

현장 문제 해결 사례 - Helium Leak Test 검사 프로세스 개선

HE LEAK TEST 검사 과정에서 누설 가능 구간을 체계적으로 관리하고,
검사 순서를 표준화하여 검사 시간과 작업 편차를 효과적으로 개선했습니다.



사용장비

-  HELIUM LEAK TEST 장비 (ASM182):
HELIUM을 이용한 기밀성 검사 장비
-  SNIFFER MODE NOZZLE (내장 액세서리):
누설 의심 구간 탐착을 위한 스니퍼 노즐
-  VACUUM GAUGE (PKR251):
진공도(기압 상태) 모니터링 게이지

주요 누설 가능 구간 예시



Weld Bead NW Flange quick clamp Centering + O-ring

문제 상황

- 1회 검사에 4시간 이상 소요
- 누설 의심 구간이 다양해 반복 확인 및 재작업 증가
- 작업자 숙련도에 따라 검사 시간 및 결과 편차 발생

원인 분석

- 누설 가능 지점이 여러 구간에 분산
- 표면 먼지, 오염 등 오염 요인이 검사 정확도에 영향
- 검사 순서와 반복 확인 기준이 표준화되어 있지 않음

개선 활동

- 검사 전 표면 오염 제거 절차 정리
- 주요 누설 가능 구간을 우선 점검 대상으로 분류
- 반복 확인 구간과 기준 명확화
- 검사 흐름 표준화로 작업자 편차 최소화

개선 결과

- 4시간 이상 → 1시간 30분 (약 60% 이상 검사 시간 단축)
- 반복 확인 및 재작업 감소, 검사 효율성과 안정성 향상

문제 해결은 감이나 숙련도에 의존하는 것이 아니라, 반복 가능한 절차와 기준을 만드는 과정이라는 점을 배웠습니다.

주요 프로젝트

HATER / Egg Studio

UNITY WEBGL 기반 웹게임 플랫폼

HATER는 UNITY WEBGL 기반 중세 디펜스 게임을 웹 브라우저에서 별도 설치 없이 실행할 수 있도록 서비스화한 웹게임 플랫폼 프로젝트입니다.

EGG STUDIO 웹 포털에서 게임 소개, 플레이 진입, 랭킹, 피드백, 뉴스 등 게임 이용 전후의 흐름을 하나의 서비스로 구성했습니다.

WEBGL 빌드 리소스를 안정적으로 제공하기 위해 S3, CLOUDFRONT, DOCKER/NGINX, GITLAB CI/CD 기반 배포 환경을 함께 구성했습니다.



6인 팀 프로젝트

FE/INFRA 1명
BACKEND 1명
UNITY CLIENT 4명

MY ROLE

FRONTEND · INFRA
웹 포털 구현
WEBGL 배포 환경 구성
CI/CD 파이프라인 구축

PERIOD

2026.04 ~ 2026.05
약 7주

주요 기능



게임 포털

게임 소개, 상세 정보, 플레이 진입 제공



웹게임 실행

UNITY WEBGL 빌드를 브라우저에서 실행



랭킹 / 기록

실시간/주간 랭킹 조회 및 상위 유저 정보 제공



피드백

게임 의견 및 개선 요청 등록



운영 콘텐츠

플레이 기록 및 사용자 정보 확인

서비스 플로우

사용자 여정 및 주요 기능 흐름

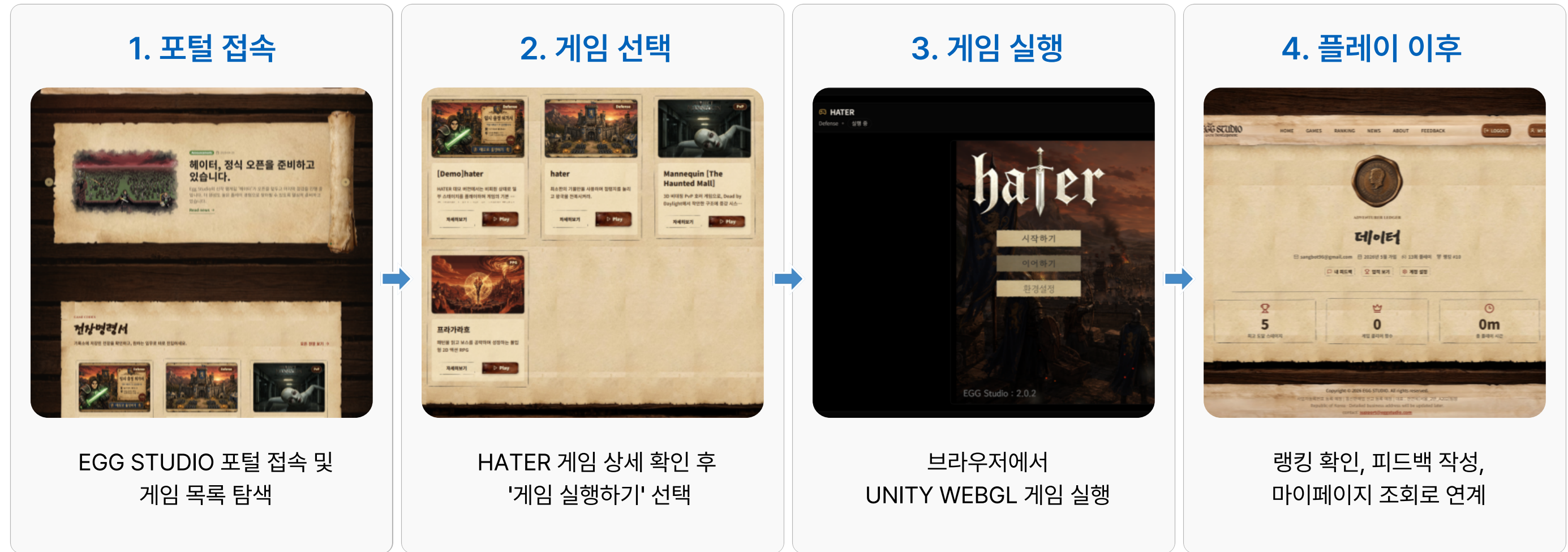
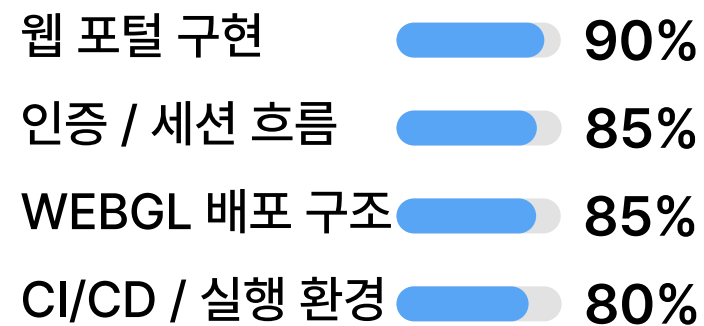
주요 흐름 요약

사용자는 회원가입 및 로그인 후 게임을 실행하고, 게임 플레이를 통해 랭킹, 피드백, 마이페이지 등 다양한 기능을 이용할 수 있습니다.

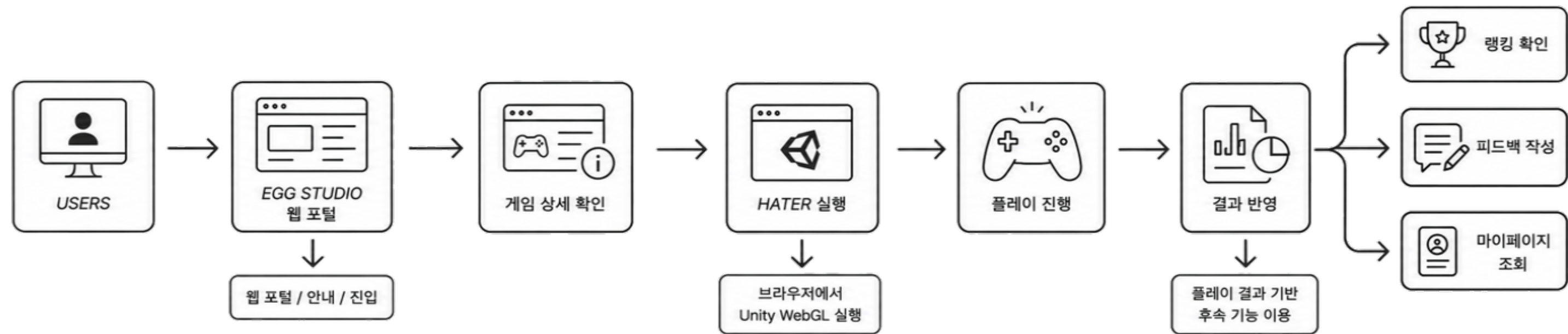
핵심 흐름

- 게임포털 탐색
- 게임 상세 확인 및 실행 진입
- UNITY WebGL 게임 플레이
- 랭킹 피드백 마이페이지 연계

기여도



USER FLOW



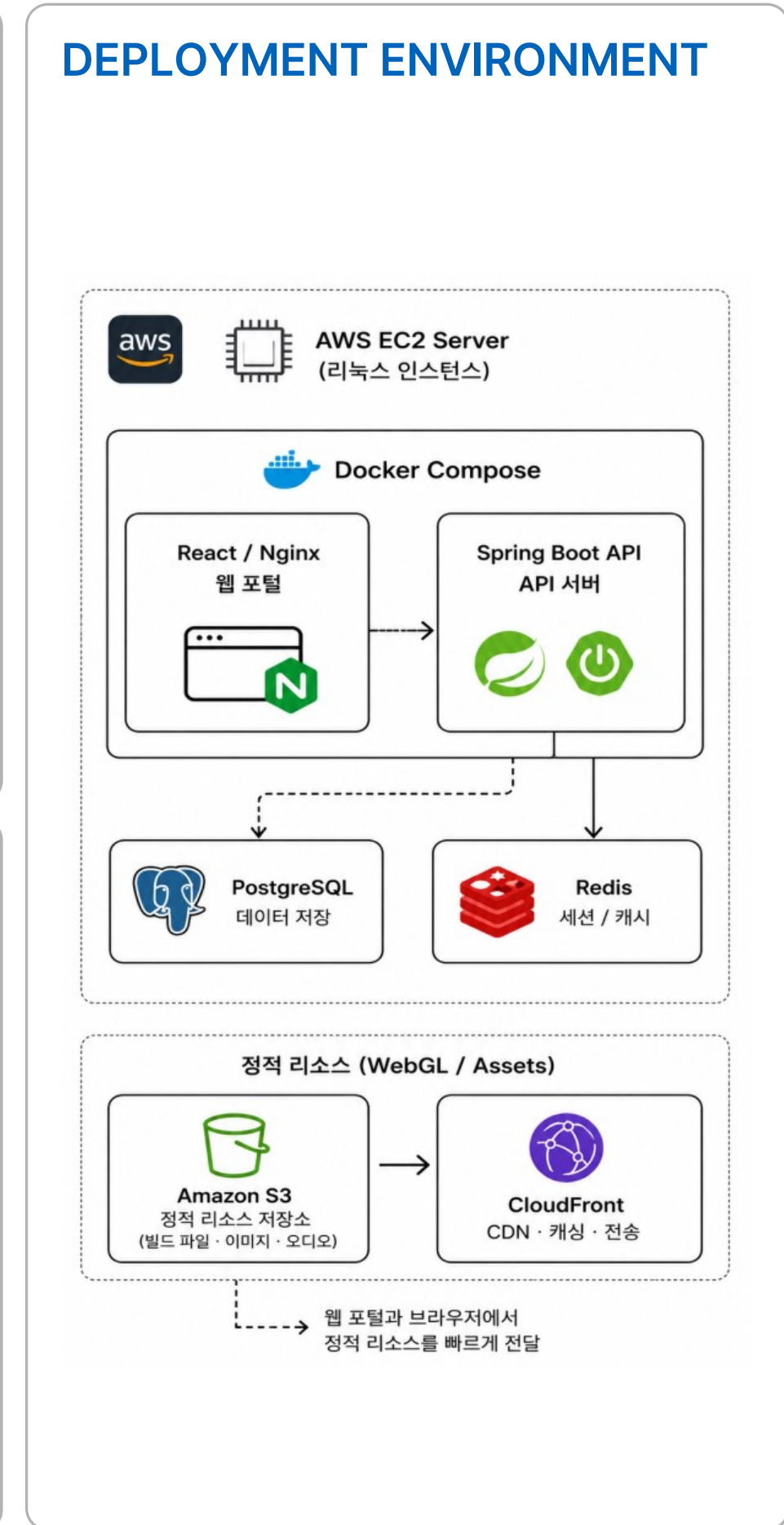
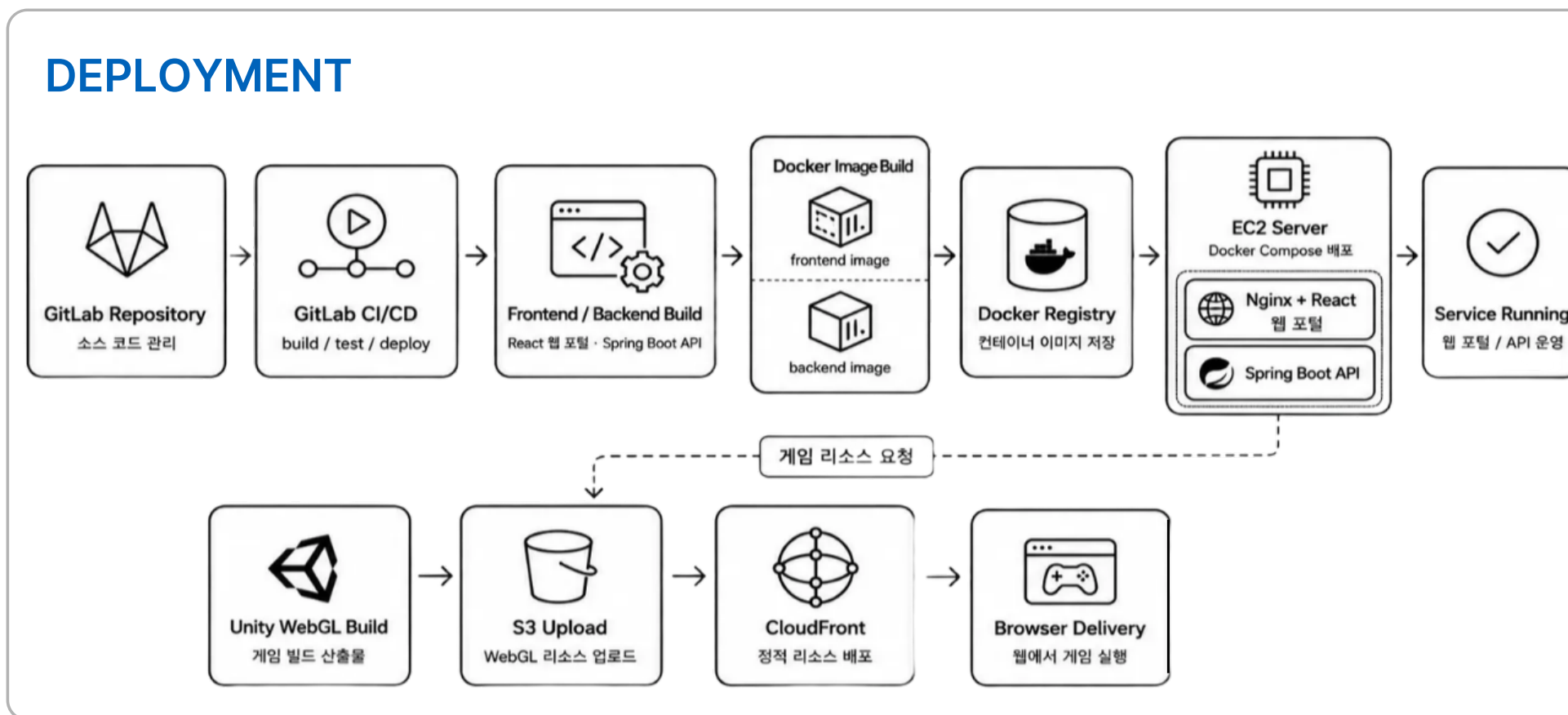
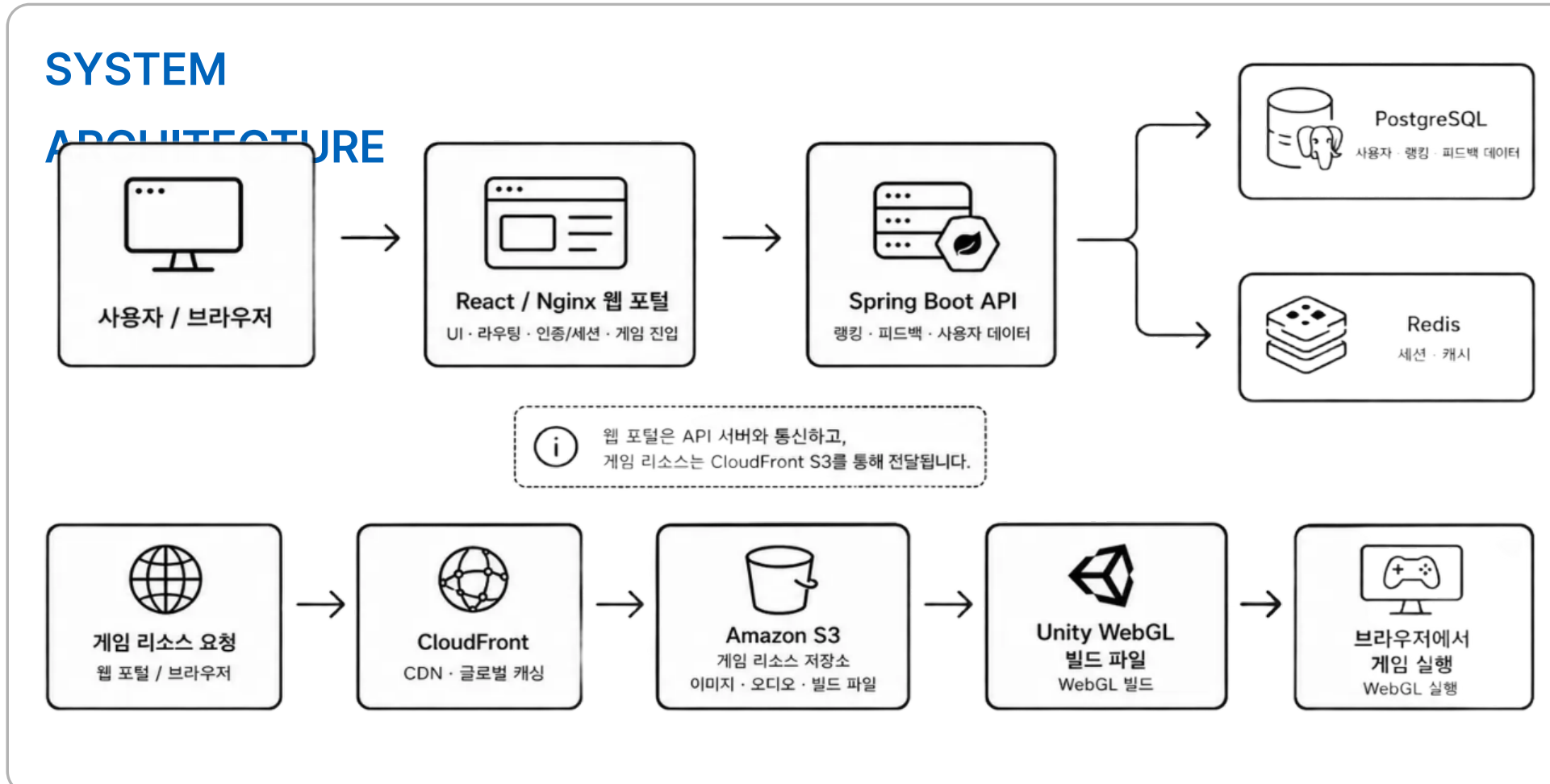
HATER / Egg Studio

아키텍처 / 배포

시스템 구조 및 배포 환경

핵심포인트

- UNITY WebGL 게임을 웹에서 안정적으로 제공
- REACT + SPRING BOOT 모듈형 구조
- S3 + CLOUDFRONT로 정적 리소스 빠르고 안정적인 제공
- DOCKER / NGINX 기반 실행 환경 분리 및 안정화
- GITLAB CI/CD를 활용한 자동 빌드 및 배포 파이프라인



문제 해결

구현 과정에서 해결한 핵심 이슈

핵심 이슈 요약

WEBGL 서비스 안정화
대용량 빌드 파일의 안정적인 전달을 위한 정적 리소스 인프라 구성

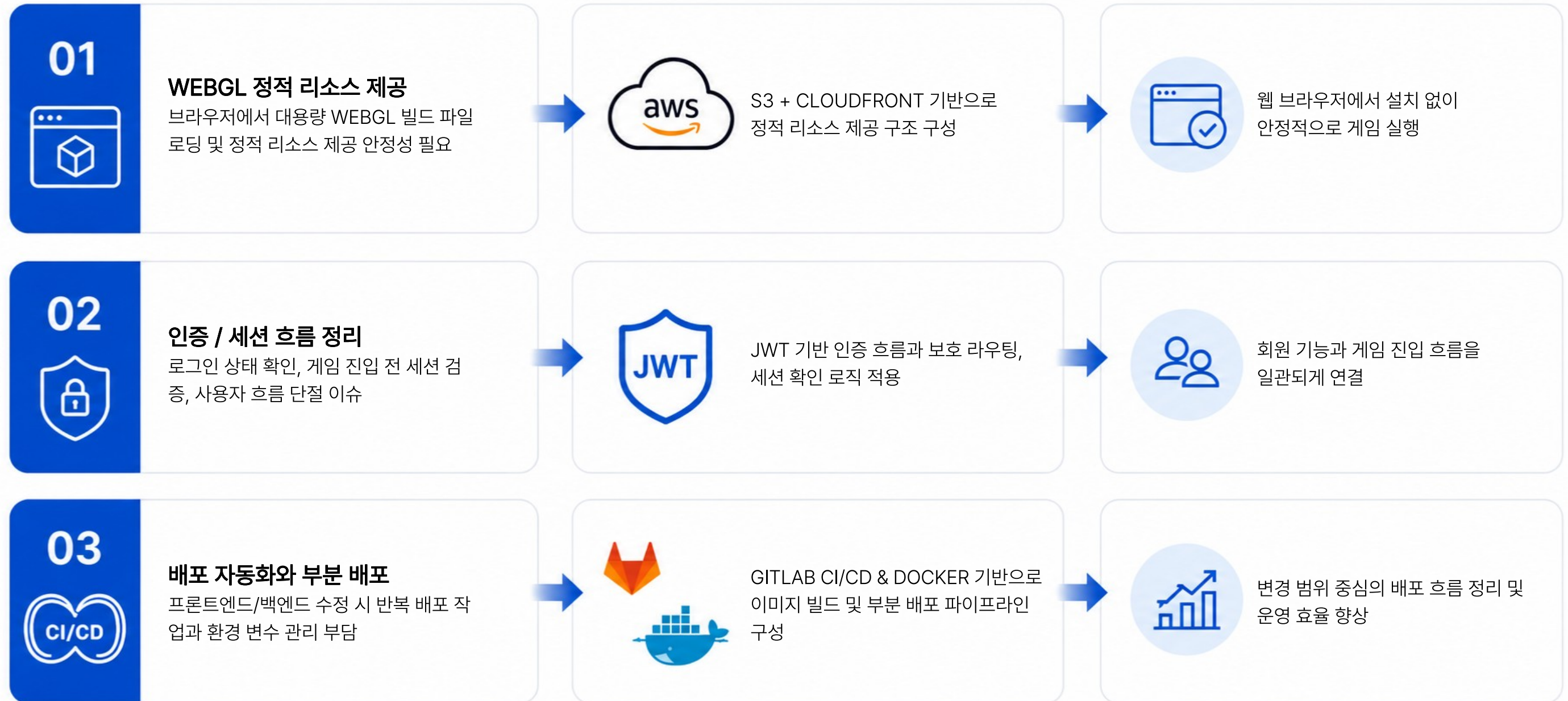
인증 및 세션 흐름 정비
JWT 기반 인증과 보호 라우팅, 세션 검증 로직으로 사용자 흐름 일관성 확보

배포 자동화 및 효율화
CI/CD와 DOCKER 기반으로 부분 배포 파이프라인을 구축해 운영 부담 최소화

PROBLEM

SOLUTION

RESULT



핵심이슈

3건

배포 흐름

자동화

WEBGL

서비스화

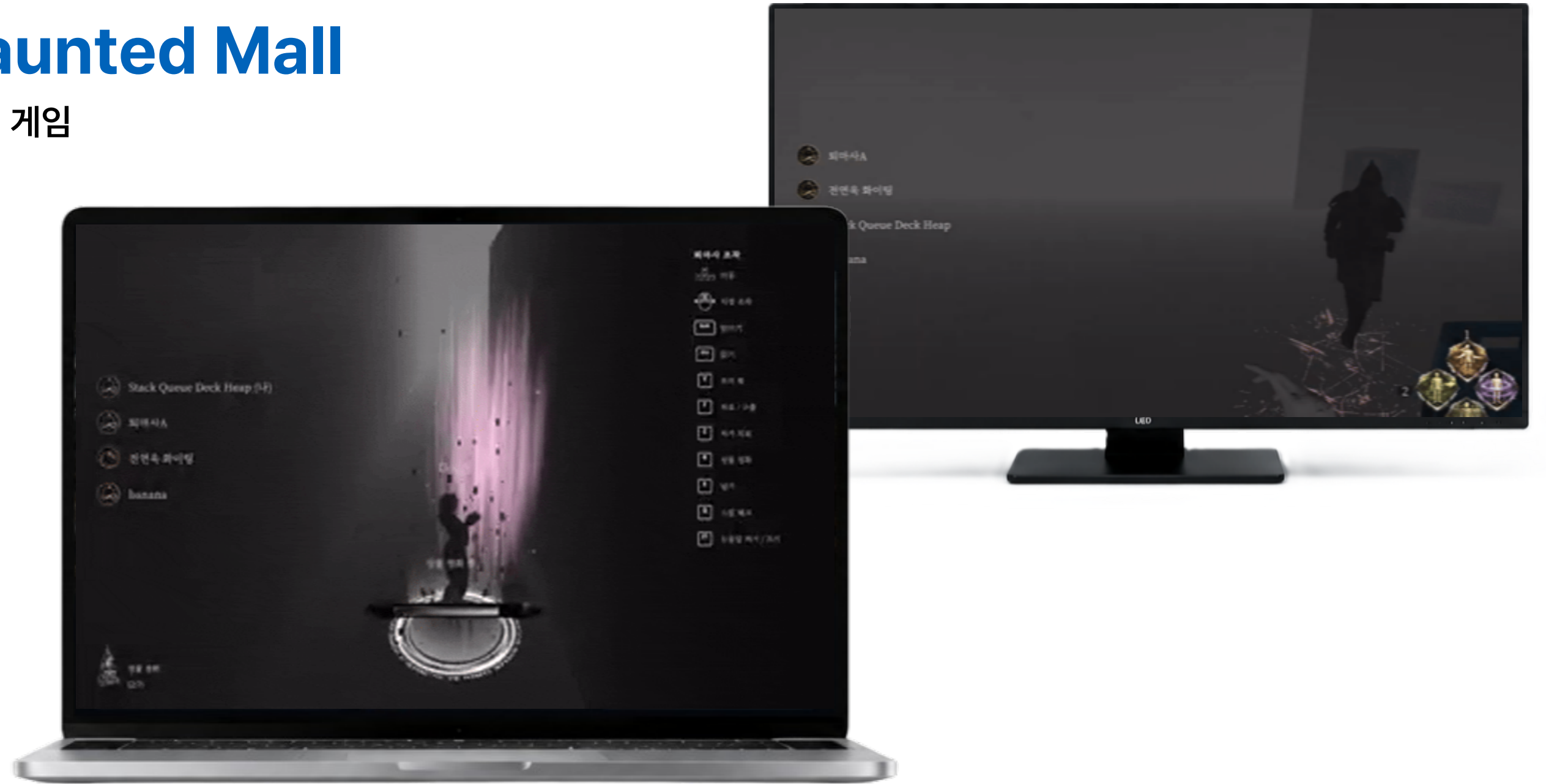
MANNEQUIN : The Haunted Mall

PHOTON FUSION 기반 3D 비대칭 PVP 호러 게임

MANNEQUIN은 폐몰을 배경으로 한 3D 비대칭 PVP 호러 게임 프로젝트입니다.

플레이어는 네트워크 기반 멀티플레이 환경에서 방을 생성하고 입장하며, 역할에 따라 서로 다른 흐름으로 게임에 진입하도록 구성했습니다.

저는 JOIN ROOM, 세션 생성·입장·종료 흐름과 역할별 로딩 씬 전환 로직을 구현했으며, GITLAB RUNNER 기반 CI 검증 환경 운영과 MATTERMOST 웹훅 기반 협업 자동화도 담당했습니다.



6인 팀 프로젝트

UNITY CLIENT 중심 개발

MY ROLE

멀티플레이 진입 흐름 구현

역할별 로딩 씬 분기

CI 검증 환경 지원

MATTERMOST 알림 자동화

PERIOD

2026.03 ~ 2026.04

약 6주

주요 기능



멀티플레이 방 입장

방 생성, 입장, 세션 참여 흐름 제공



비대칭 PVP 플레이

역할에 따라 다른 목표와 플레이 방식 제공



역할별 게임 진입

로딩 씬과 시작 흐름을 역할별로 분기



네트워크 플레이

PHOTON FUSION 실시간 멀티플레이



호러 맵 탐색

밀폐된 공간에서 추적·회피 중심 플레이 제공

서비스 플로우

사용자 여정 및 주요 기능 흐름

주요 흐름 요약

플레이어는 로비에서 방을 생성하거나 입장한 뒤, PHOTON FUSION 기반 세션에 참여합니다.

이후 역할에 따라 서로 다른 로딩 흐름을 거쳐 게임에 진입합니다.

핵심 흐름

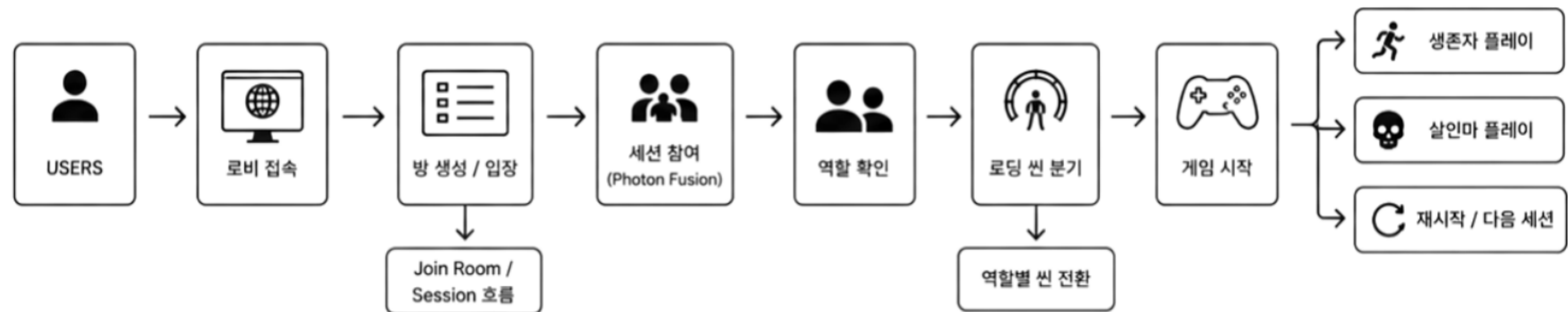
- 로비 접속 및 방 목록 확인
- 방 생성 / 입장 및 세션 참여
- 역할 분기 및 로딩 씬 전환
게임 시작 및 플레이 진행

기여도

- 멀티플레이 진입 흐름 80%
- 역할별 로딩 씬 분기 80%
- CI / 협업 환경 지원 75%
- 최적화 / 검증 지원 70%



USER FLOW



MANNEQUIN : The Haunted Mall

역할 및 주요 기여

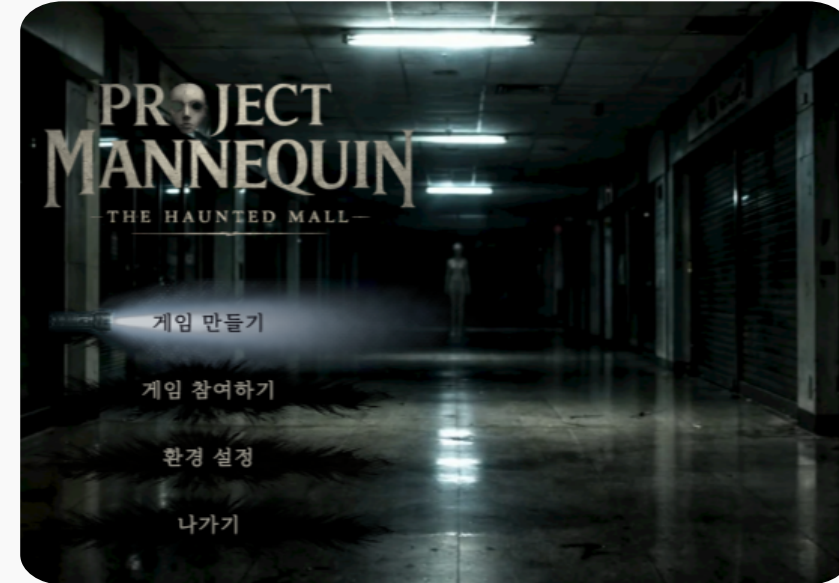
멀티플레이 진입 흐름 및 지원 영역

담당역할

- PHOTON FUSION 기반 멀티플레이 진입 흐름 및 역할 분기 구현
- 로비 채팅 시스템과 UI 에셋 적용
- CI / 협업 환경 지원 및 테스트

1. 게임 진입 흐름 구현

- 로비, 방 목록, 입장, 세션 참여 로직 구현
- PHOTON FUSION 기반 세션 관리 및 동기화
- 게임 시작 전 사용자 흐름 연결



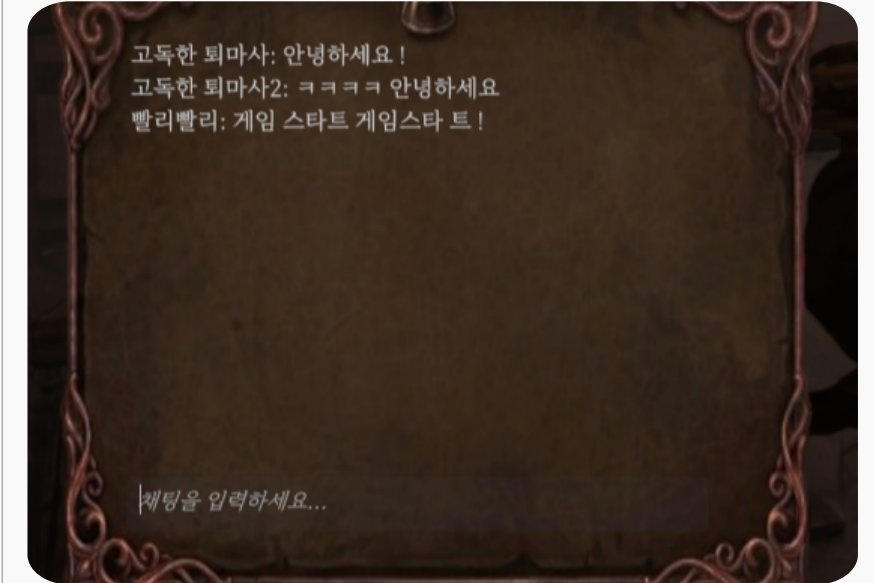
2. 역할별 로딩 씬 분기

- 생존자 / 추적자 역할 선택 상태 전달
- 역할에 따라 다른 로딩 씬 및 진입 흐름 분기
- 씬 전환 시 상태 정보 연계



3. 로비 채팅 시스템 구현

- PHOTON 기반 로비 채팅창 구현
- 입장 유저 간 메시지 송수신 처리
- 채팅 UI와 네트워크 이벤트 연동



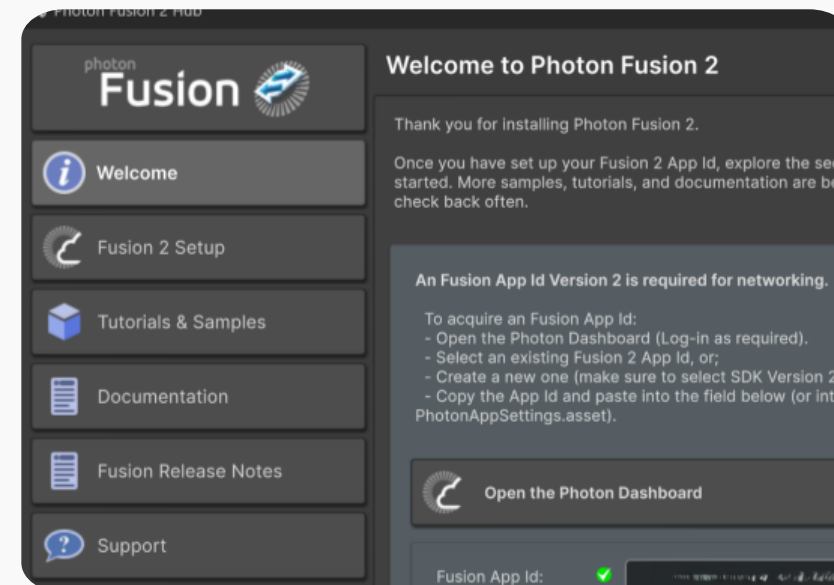
4. UI 에셋 제작 및 적용

- UI용 이미지 에셋 제작 및 적용
- 로비 / 메뉴 / 상태 표시 UI 구성 보조
- 화면 분위기에 맞는 시각 요소 정리



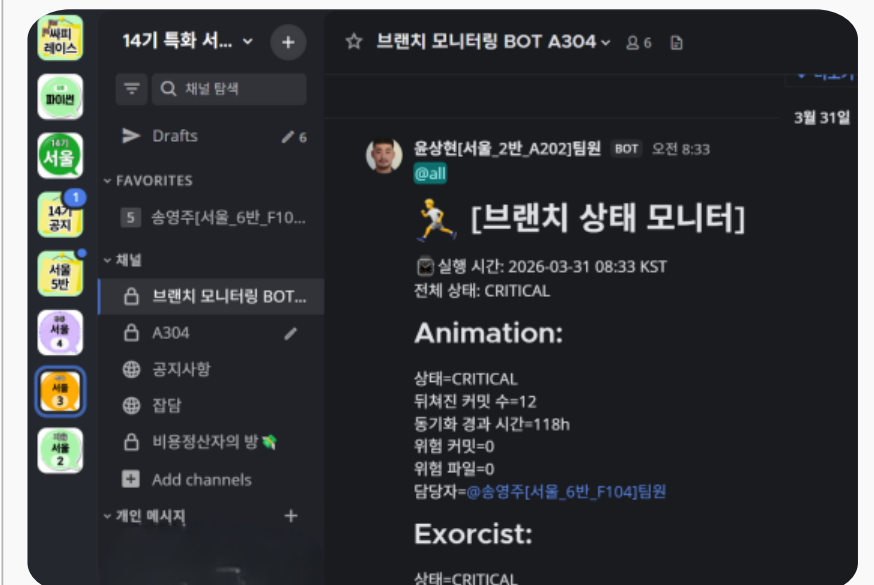
5. PHOTON FUSION 네트워크

- 멀티플레이 테스트 및 흐름 검증
- 룸 상태와 세션 연결 확인
- 플레이 진행 안정성 점검



6. CI / 협업 환경 지원


- GITLAB RUNNER 기반 CI 검증 환경 운영
- MATTERMOST 웹훅 기반 협업 자동화 지원
- 이슈 공유 및 빌드 확인 프로세스 정리



문제 해결


멀티 플레이 환경에서 해결한 핵심 이슈

🎯 핵심 이슈 요약




역할별 진입 흐름 분리

생존자와 추적자의 진입 로직을 분리해 역할별 초기화와 시작 흐름을 안정적으로 구성



상호작용 상태 동기화

행거, 성유물 등 상호작용 이벤트가 모든 클라이언트에서 일관되게 반영되도록 구조 정리



멀티플레이 최적화

네트워크 트래픽, 스폰 비용, 렌더링 부담을 줄여 플레이 경험을 개선



핵심이슈	멀티 플레이	협업 / UX
4건	흐름 안정화	개선

DDIP

실시간 좌석 예매 및 좌석 교환 서비스

DDIP는 실시간 좌석 예매와 좌석 교환을 제공하는 웹 서비스 프로젝트입니다.

서비스는 MSA 구조를 기반으로 예매, 좌석, 교환 등 주요 도메인을 분리하고, KAFKA를 활용한 비동기 이벤트 처리와 REDIS 기반 상태 관리를 통해 실시간 좌석 흐름을 안정적으로 처리하도록 구성되었습니다.

저는 프론트엔드 담당으로 공연 탐색, 좌석 선택, 예매 진행, 모바일 티켓, 좌석 교환 화면을 구현하며, 분리된 API 응답과 상태 변화를 사용자 흐름에 맞게 연결하는 데 집중했습니다.

6인 팀 프로젝트

FRONTEND 2명
BACKEND 3명
INFRA 1명

MY ROLE

FRONTEND
예매 화면 UI 구현
좌석 선택 흐름 설계
교환 요청 화면 구현

PERIOD

2026.01 ~ 2026.02
약 6주



주요 기능



공연 탐색
공연 목록 및 상세 정보 확인



실시간 좌석 선택
좌석 상태를 반영한 예매 흐름 제공



예매 진행
회차 선택부터 결제 전 단계까지 연결



모바일 티켓
예매 내역과 QR 티켓 확인



좌석 교환
교환 요청 및 매칭 상태 확인

서비스 플로우

예매부터 좌석 교환의 사용자 여정

주요 흐름 요약

공연 탐색, 좌석 선택, 결제, 모바일 티켓 입장까지 이어지는 예매 흐름과 좌석 교환 요청·매칭·결과 반영 과정을 하나의 사용자 여정으로 구성했습니다.

실시간 좌석 및 교환 상태는 REDIS·KAFKA 기반 처리 구조와 연계되어 빠르게 반영됩니다.

핵심 흐름

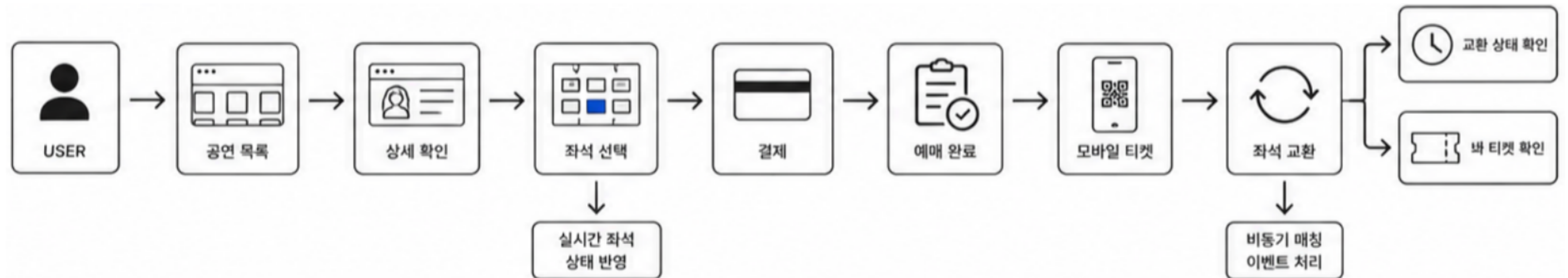
- 공연 탐색 및 상세 확인
- 좌석/구역 선택
- 결제 및 예매 확정
- 모바일 티켓 / QR 확인
- 좌석 교환 요청 및 매칭 확인

기여도

- 예매 UI 흐름 구현 90%
- 좌석 선택 상태 연동 85%
- 모바일 티켓 화면 구성 80%
- 좌석 교환 UX 반영 80%



USER FLOW



서비스 아키텍처

MSA기반 시스템 구조 이해

핵심 포인트

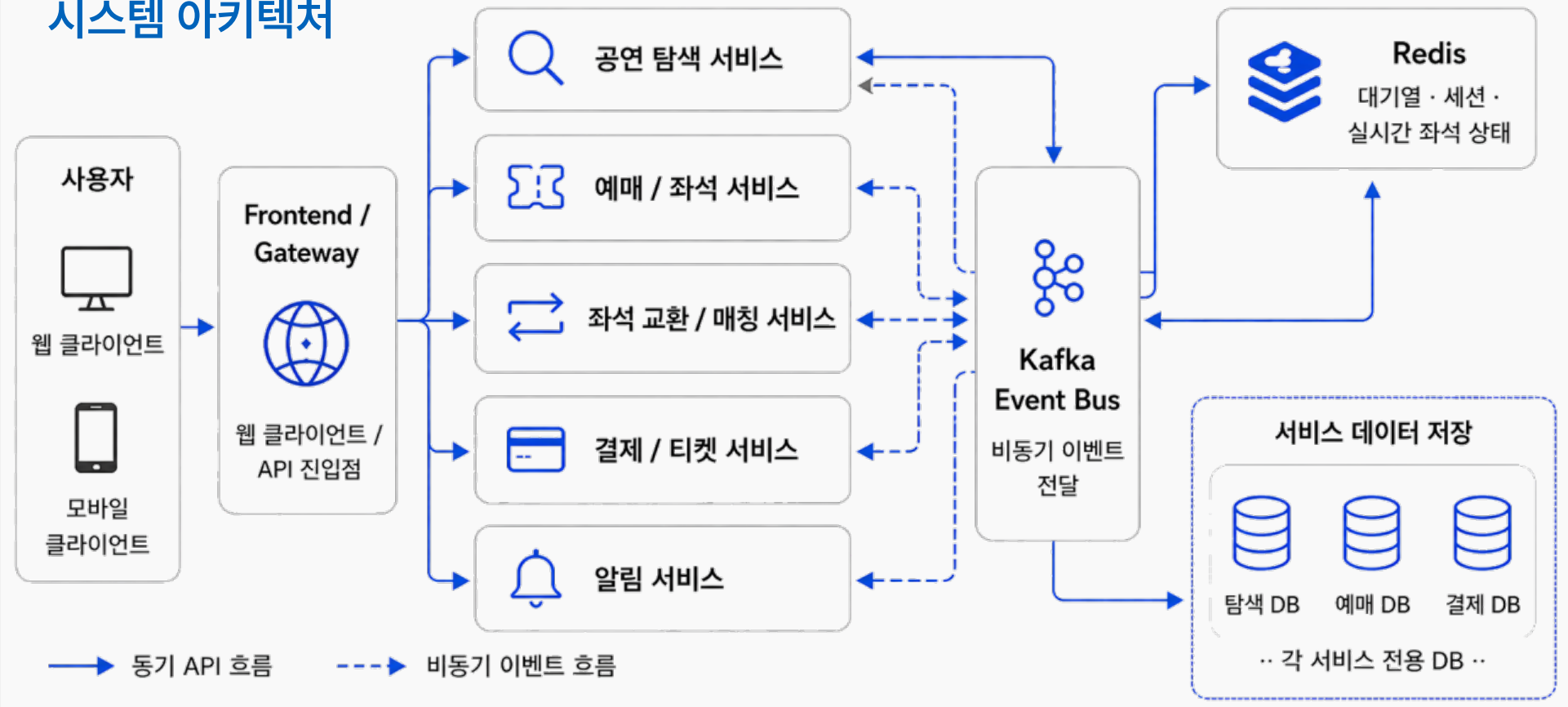
MSA 구조
기능 단위 서비스 분리로
독립 배포와 확장 가능

KAFKA 비동기 처리
예매·교환·알림 이벤트를
비동기 흐름으로 전달

REDIS 실시간 상태 반영
대기열·세션·좌석 상태를
빠르게 조회·반영

KUBERNETES 기반 확장 구조
서비스별 복제와
트래픽 분산이 가능한 운영 구조

시스템 아키텍처



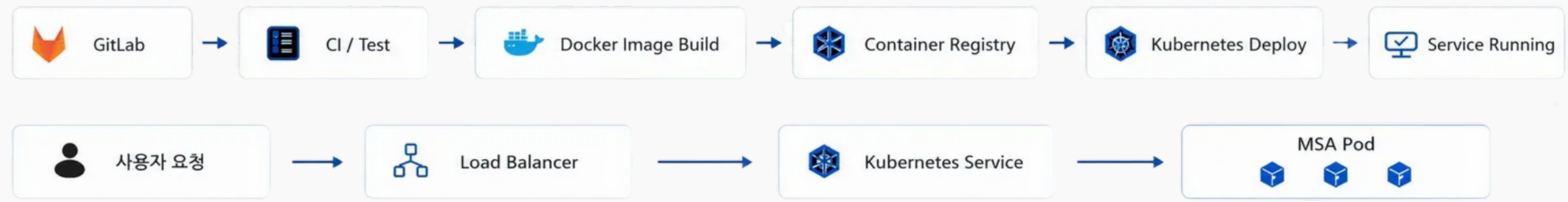
① 동기 요청은 API를 통해 처리하고, 비동기 이벤트는 KAFKA로 전달합니다.

요약

DDIP는 MSA 구조를 기반으로 기능을 서비스 단위로 분리하고, KAFKA와 REDIS를 활용해 동기·비동기 흐름을 나누어 처리한 서비스입니다.

또한 KUBERNETES 기반 복제와 확장 구조를 통해 실시간 예매 및 좌석 교환 서비스의 안정성을 고려한 구성을 갖추었습니다.

배포 흐름



성장 방향 및 마무리

현장 경험과 프로젝트를 통해 다져온 개발자로서의 방향성



현장 경험과 개발 역량의 연결

반도체 장비 엔지니어로 근무하며 문제를 빠르게 파악하고 원인을 추적하는 습관을 익혔습니다.

이러한 현장 중심의 사고는 개발 과정에서도 요구사항을 구조적으로 정리하고, 실제 사용자 흐름과 운영 관점까지 고려하는 방식으로 이어졌습니다.



프로젝트를 통해 배운 점

웹 서비스, 게임, 인프라 프로젝트를 경험하며 기능 구현 자체보다 서비스 전체 흐름을 이해하는 것이 중요하다는 점을 배웠습니다.

협업, 배포, 운영, 사용자 경험까지 연결해서 보는 시야를 키웠고, 맡은 영역을 끝까지 책임지는 태도의 중요성도 체감했습니다.



앞으로의 개발 방향

앞으로는 프론트엔드와 인프라 경험을 바탕으로 서비스 구조를 이해하는 개발자로 성장하고 싶습니다.

단순히 기능을 만드는데 그치지 않고, 안정적인 운영과 더 나은 사용자 경험까지 함께 고민할 수 있는 개발자가 되는 것이 목표입니다.

핵심 키워드



문제 해결



사용자 흐름



협업



운영 관점



지속 성장

현장을 이해하는 시선과 개발 경험을 연결하여, 더 나은 서비스를 만드는 개발자로 성장하겠습니다.