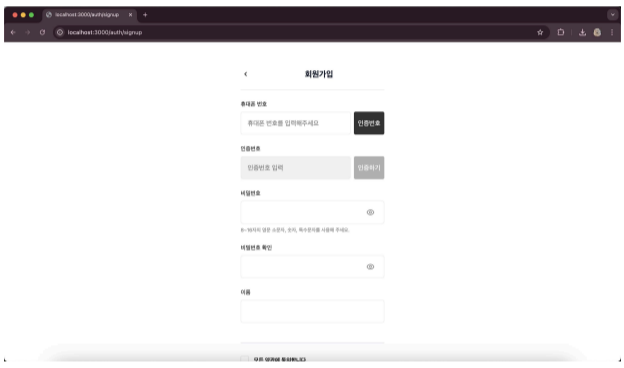
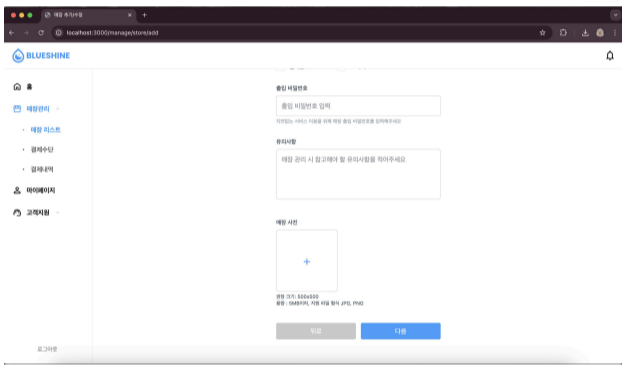
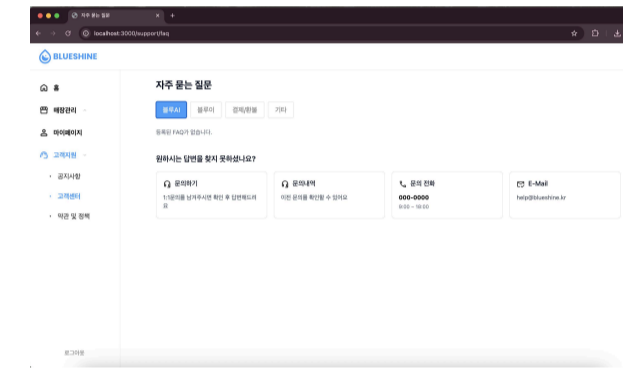


# 무인 매장 관리 플랫폼

실제 운영을 목표로 설계한 인증·결제 중심 B2B SaaS 플랫폼

사이트로 이동

<Service> 4개 도메인으로 나누어 프론트 Next.js / 백엔드 Spring Boot 코드 관리

Auth	Store	Support	Purchase
<ul style="list-style-type: none"> <li>SMS 인증 기반 회원가입/로그인</li> <li>Redis TTL로 인증 코드 관리</li> <li>인증 실패/재시도 제한</li> </ul>	<ul style="list-style-type: none"> <li>매장 등록 및 기본 정보 관리</li> <li>FE 및 매장/기기/계약 DB 설계</li> <li>AI/분석 연동 고려</li> </ul>	<ul style="list-style-type: none"> <li>게시판, 문의하기, 서비스 정책</li> <li>관리자 페이지 확장 고려 설계</li> </ul>	
			

## <Stacks>

### FE

- Framework: Next.js + Typescript
- CICD: Git Action, Vercel (CDN)
- UI 컴포넌트화, React Query, Zustand, MVVM 구조

### BE

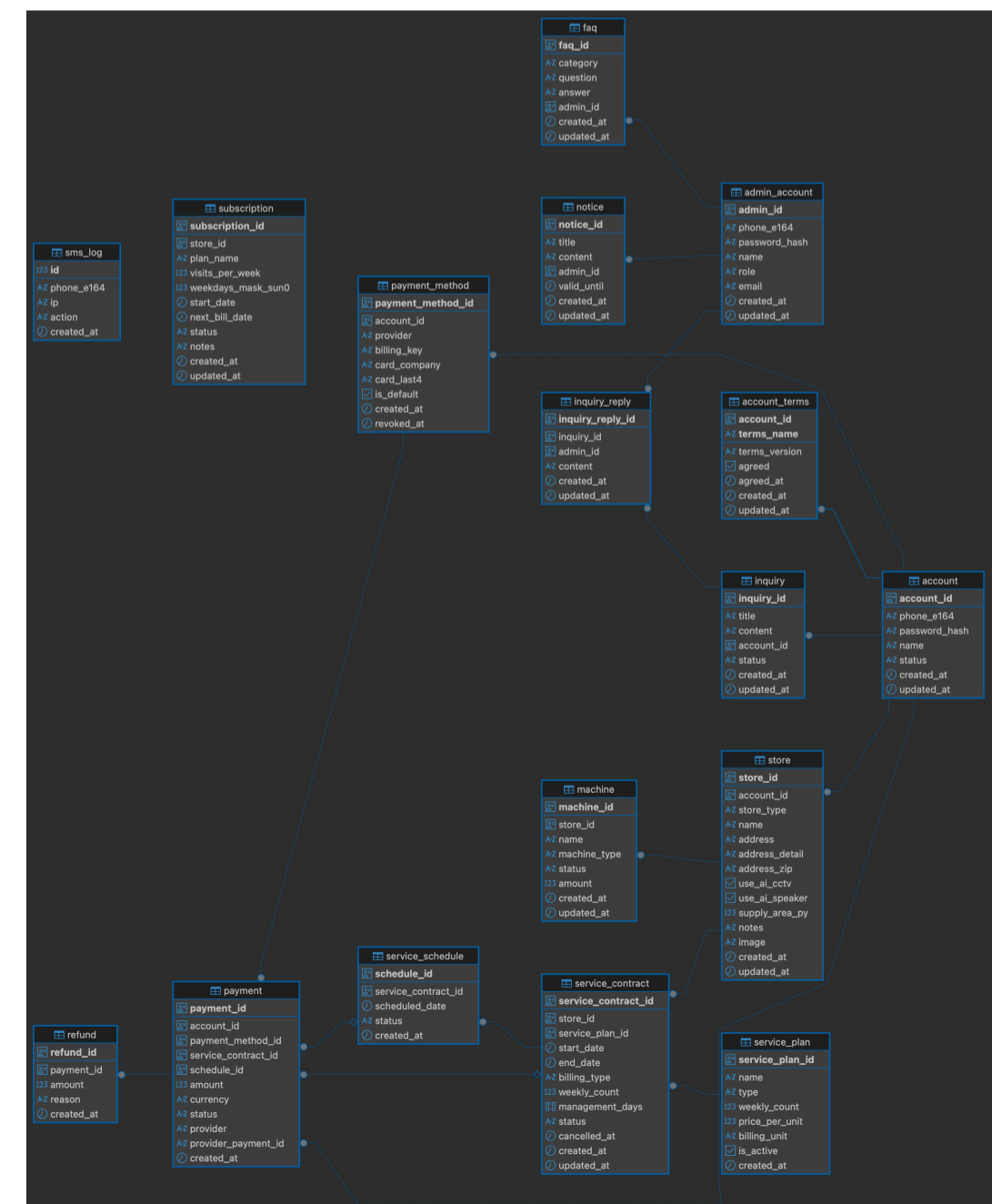
- Framework: Spring Boot + Java
- CICD: Jenkins, Git, Docker on EC2
- Exception 커스텀, MVC 패턴, API 인터페이스 객체, 롤링 업데이트

### Data

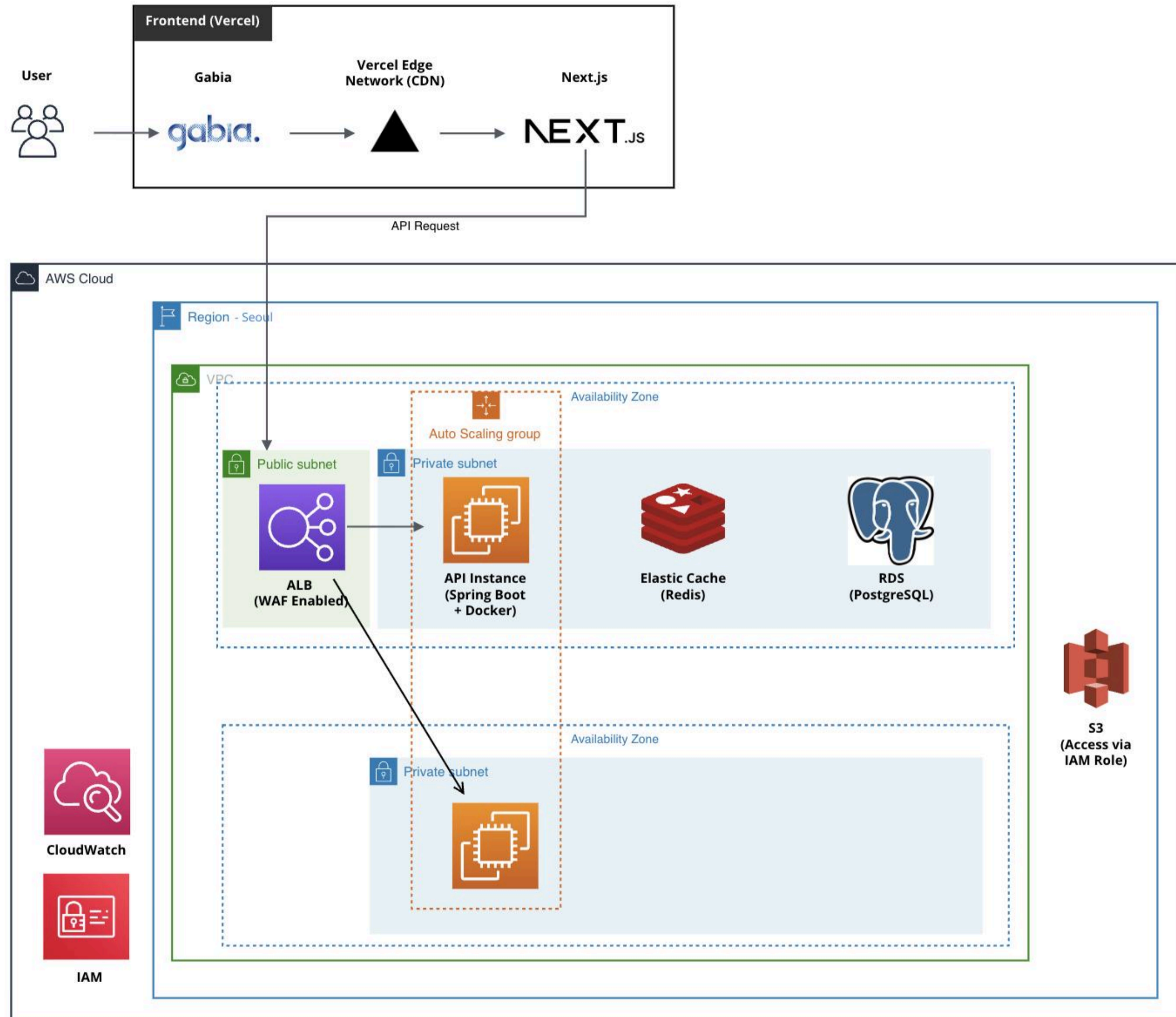
- RDS (PostgreSQL) → 데이터 정합성
- ElastiCache (Redis) → 인증 TTL, 캐시로 DB 부하 감소
- S3 → 이미지 및 정적 자산 저장

### Infra

- AWS → ALB, Auto Scaling Group, WAF



# <Infra Architecture>



## Request Flow

1. User → Vercel Edge (Next.js)
2. API 요청 → ALB (WAF)
3. EC2 ASG → Redis / RDS / S3

## Why

- CDN 중복 방지
- Private Subnet으로 보안 강화
- 관리형 DB로 안정성 확보

## Security

- WAF at ALB
- Private Subnet
- IAM Role 기반 S3 접근

## Stability

- Auto Scaling Group
- Rolling Deployment
- CloudWatch 모니터링

## Scalability

- 현재는 비용절감을 위해 단일 EC2만 사용, 일정 트래픽 이상 ASG 설정
- 쿠버네티스 기반 마이크로서비스로의 확장 고려