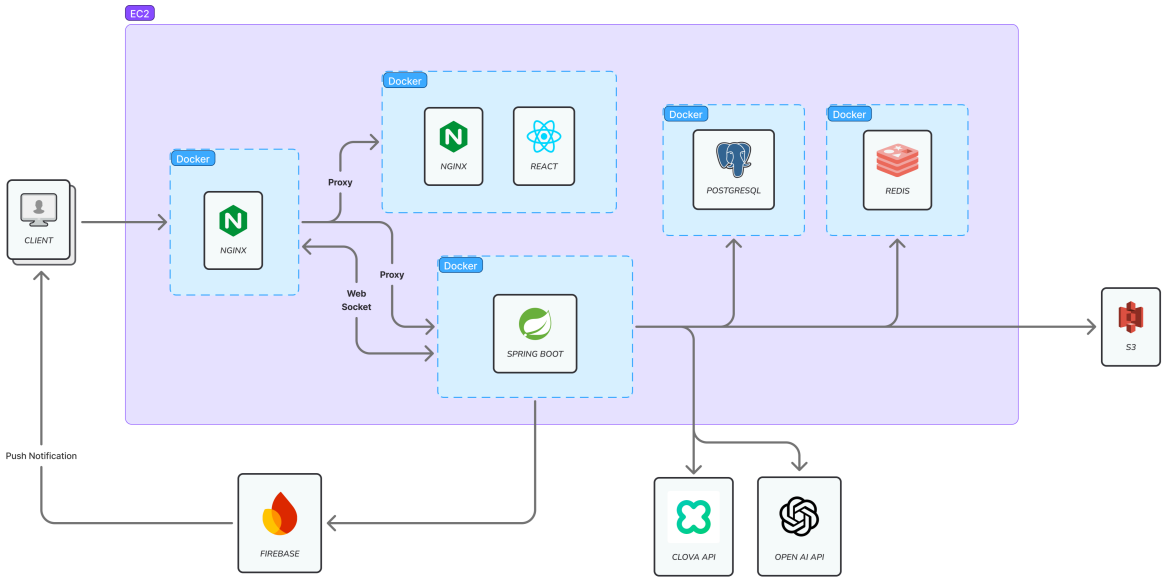
포팅 매뉴얼

> **서비스명 : 출퇴근길에 굽는 지식 한 조각, 암기빵**
> **팀명 : A704**

1. Architecture



2. Infra

VERSION

- **Domain** : remembread.co.kr
- **Docker** : 27.5.1
- **Docker compose** : v2.32.4
- **Nginx** : 1.23-alpine
- **Certbot** : 최신
- **SSL 프로토콜** : TLSv1.2, TLSv1.3

주요 포트 번호

컴포넌트	포트 번호
Nginx	80, 443
React	3000
Spring Boot	8080
PostgreSQL	5432
Redis	6379
Prometheus	9090
Grafana	4000

실행 가이드

1. `docker-compose.yml` 파일 작성

```
version: '3.9'

services:
  ############################################################
  # 1. Nginx Reverse Proxy
  ############################################################
  nginx-proxy:
    build:
      context: ./nginx
```

```
    dockerfile: Dockerfile
  container_name: nginx-proxy
  restart: always
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - /etc/letsencrypt:/etc/nginx/certs
    - nginx-html:/usr/share/nginx/html
  networks:
    - app-network

############################################################
# 2. Certbot (별도 컨테이너로 인증서 갱신)
############################################################
certbot:
  image: certbot/certbot
  container_name: certbot
  volumes:
    - nginx-html:/usr/share/nginx/html
    - nginx-certs:/etc/nginx/certs
  entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 12h
  networks:
    - app-network

############################################################
# 3. PostgreSQL
############################################################
postgres:
  image: postgis/postgis:15-3.3
  container_name: postgres
  restart: always
  environment:
    POSTGRES_USER: ssafy
    POSTGRES_PASSWORD: bradpitt704
    POSTGRES_DB: postgres
  volumes:
    - postgres-data:/var/lib/postgresql/data
```

```
  ports:
    - "127.0.0.1:5432:5432"
  networks:
    - app-network

#############################################################
# 4. Redis
#############################################################
redis:
  image: redis:7
  container_name: redis
  restart: always
  ports:
    - "127.0.0.1:6379:6379"
  volumes:
    - redis-data:/data
  command: sh -c "chown -R redis:redis /data && redis-server"
  networks:
    - app-network

#############################################################
# 5. Prometheus
#############################################################
prometheus:
  image: prom/prometheus:latest
  container_name: prometheus
  ports:
    - "127.0.0.1:9090:9090"
  volumes:
    - ./prometheus.yml:/etc/prometheus/prometheus.yml
    - ./prometheus-data:/prometheus
  networks:
    - app-network

#############################################################
# 6. Grafana
#############################################################
grafana:
```

```
    image: grafana/grafana:latest
    container_name: grafana
    ports:
      - "4000:3000"
    volumes:
      - ./grafana-data:/var/lib/grafana
    depends_on:
      - prometheus
    networks:
      - app-network

volumes:
  nginx-html:
  nginx-certs:
  postgres-data:
  redis-data:

networks:
  app-network:
    driver: bridge
```

2. **폴더 이동**

```
cd nginx
```

3. `Dockerfile` **파일 작성**

```
FROM nginx:1.23-alpine

# 커스텀 Nginx 설정 파일 복사 (default.conf)
# 이 설정 파일에서 /api와 /를 각각 다른 백엔드로 프록시하는 설정을 포함시킵니다.
COPY default.conf /etc/nginx/conf.d/default.conf

# 인증서 파일은 공유 볼륨을 통해 주입됨 (예: /etc/nginx/certs)
VOLUME ["/etc/nginx/certs"]

EXPOSE 80 443
```

```
CMD ["nginx", "-g", "daemon off;"]
```

4. `default.conf` **파일 작성**

```
server {
    listen 80;
    server_name k12a704.p.ssafy.io remembread.co.kr www.remembread.co.kr

    client_max_body_size 50M;

    location ^~ /.well-known/acme-challenge/ {
        root /usr/share/nginx/html;
        default_type "text/plain";
        try_files $uri =404;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl;
    server_name k12a704.p.ssafy.io remembread.co.kr www.remembread.co.kr

    client_max_body_size 50M;

    ssl_certificate /etc/nginx/certs/live/k12a704.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/nginx/certs/live/k12a704.p.ssafy.io/privkey.pem;
    ssl_session_cache shared:SSL:10m;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    location ^~ /.well-known/acme-challenge/ {
        root /usr/share/nginx/html;
        default_type "text/plain";
        try_files $uri =404;
```

```
    }

    location /api/ {
        proxy_pass http://k12a704.p.ssafy.io:8080/;

        proxy_http_version 1.1;
        proxy_set_header Connection '';

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_buffering off;
        proxy_cache off;
        chunked_transfer_encoding on;
        proxy_read_timeout 3600s;
        proxy_send_timeout 3600s;
    }

    location / {
        proxy_pass http://k12a704.p.ssafy.io:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

5. **폴더 이동**

```
cd ..
```

6. **도커 실행**

```
docker compose up -d --build
```

7. **Certbot을 통해 최초 인증서 발급**

> 인증서는 최초에 한 번만 수동으로 발급받고, 이후 `certbot` 컨테이너에서 자동 갱신됨

```
docker run --rm \
  -v nginx-html:/usr/share/nginx/html \
  -v nginx-certs:/etc/nginx/certs \
  certbot/certbot certonly --webroot \
  -w /usr/share/nginx/html \
  -d remembread.co.kr \
  --email your@email.com \
  --agree-tos \
  --no-eff-email
```

8. **Nginx 재실행**

```
docker restart nginx-proxy
```

3. Frontend

VERSION

- **nodeJs** : 22
- **Vite** : 6.3.1
- **React** : 19.0.0
- **Typescript** : 5.7.2
- **React Router** : 7.4.0
- **Tailwind CSS** : 3.4.17
- **Zustand** : 5.0.3
- **Axios** : 1.9.0

실행 가이드

1. **Git clone (fe 브랜치만)**

```
git clone -b fe --single-branch https://lab.ssafy.com/s12-final/S12P31A704.git
```

2. **폴더 이동**

```
cd RememBread_frontend
```

3. `.env` **파일 작성**

```
# 백엔드 API 요청 링크
# VITE_APP_BASE_URL=https://k12a704.p.ssafy.io/api
VITE_APP_BASE_URL=https://remembread.co.kr/api

# 소셜 로그인용 클라이언트 ID (kakao, naver, google)
VITE_KAKAO_CLIENT_ID=fa3d810c08ba4120bb3294a94f7696d7
VITE_NAVER_CLIENT_ID=t5BYZ3J2kAlf3socHE7f
VITE_GOOGLE_CLIENT_ID=365476883445-led6dhq6oi5fnsjmnnefccacen9ob

# 프론트엔드 배포 링크
VITE_FRONT_BASE_URL=http://localhost:5173
VITE_FRONT_BASE_URL=https://remembread.co.kr/

# 지도 CLIENT_ID
VITE_CLIENT_ID = 00w9px3mo8
```

4. **의존성 설치**

```
npm install
```

5. **빌드**

```
npm run build
```

6. **도커 이미지 빌드**

```
docker build -t my-frontend .
```

7. **컨테이너 실행**

```
docker run -d --name my-frontend -p 3000:3000 my-frontend
```

4. Backend

VERSION

- **Java Version** : 17 (Configured using Java Toolchain)
- **Spring Boot Version** : 3.4.3
- **Dependency Management Version** : 1.1.7
- **Database** : PostgreSQL (Using org.postgresql:postgresql)
- **ORM** : Spring Data JPA 3.2.3 with Hypersistence Utils, Hibernate Spatial
- **Security** : JWT (io.jsonwebtoken:jjwt)
- **Caching & Storage** : Redis
- **API Documentation** : SpringDoc OpenAPI 2.8.5
- **Web** : Spring Boot Web
- **Testing Frameworks** : JUnit

실행 가이드

1. **Git clone (be 브랜치만)**

```
git clone -b be --single-branch https://lab.ssafy.com/s12-final/S12P31A704.git
```

2. **폴더 이동**

```
cd be/remembread
```

3. `.env` **파일 작성**

```
POSTGRESQL_URL=localhost:5432
POSTGRESQL_USERNAME=postgres
POSTGRESQL_PASSWORD=ssafy

REDIS_HOST=localhost
```

```
JWT_SECRET_KEY=nHpOdfNXXR+PrE5Y1xhvtuX6zCsgvbwGYNyhCKpjhiE=

KAKAO_CLIENT_ID=fa3d810c08ba4120bb3294a94f7696d7
KAKAO_CLIENT_SECRET=BBXOWywvQ4jie96VxdJPiHG4prGuT44l
KAKAO_REDIRECT_URL=http://localhost:8080/auth/login/kakao

NAVER_CLIENT_ID=t5BYZ3J2kAlf3socHE7f
NAVER_CLIENT_SECRET=PlcFBHx4x_
NAVER_REDIRECT_URL=http://localhost:8080/auth/login/naver

GOOGLE_CLIENT_ID=365476883445-led6dhq6oi5fnsjmnnefccacen9obar1.ap
GOOGLE_CLIENT_SECRET=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
GOOGLE_REDIRECT_URL=http://localhost:8080/auth/login/google

OPENAI_API_KEY=sk-proj-fRC0pB17qFukWmhw9qXmIJ_7KZKEYEugHMyfXPR

CLOVA_OCR_URL=https://0317heil2b.apigw.ntruss.com/custom/v1/41233/ad2f
CLOVA_OCR_KEY=QkZERGJETXNuV2JmWGlMbWRCeGtmT3J4RXN2ZVprem4

AWS_S3_BUCKET=remembread-bucket
AWS_ACCESS_KEY_ID=AKIAQJDAHDMY34BYF5KW
AWS_SECRET_ACCESS_KEY=TMDs8QJbSWGnYsqNA7YUUS0aTE8X90+4DDe

FCM_BUCKET=remembread-secret-bucket
FCM_KEY=remembread-firebase-adminsdk-fbsvc-39b7cf669b.json
```

4. **Gradle Wrapper에 실행 권한 부여**

```
chmod +x gradlew
```

5. **빌드**

```
./gradlew clean build
```

6. **실행**

```
java -jar build/libs/*.jar
```

5. Database

VERSION

- **PostgreSQL** : 17.4
- **PostGIS** : 3.5.1
- **Redis** : 7.1.0

실행 가이드

1. **PostGIS 설치**

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

2. **PostgreSQL에** `remembread.dump` **파일 import**