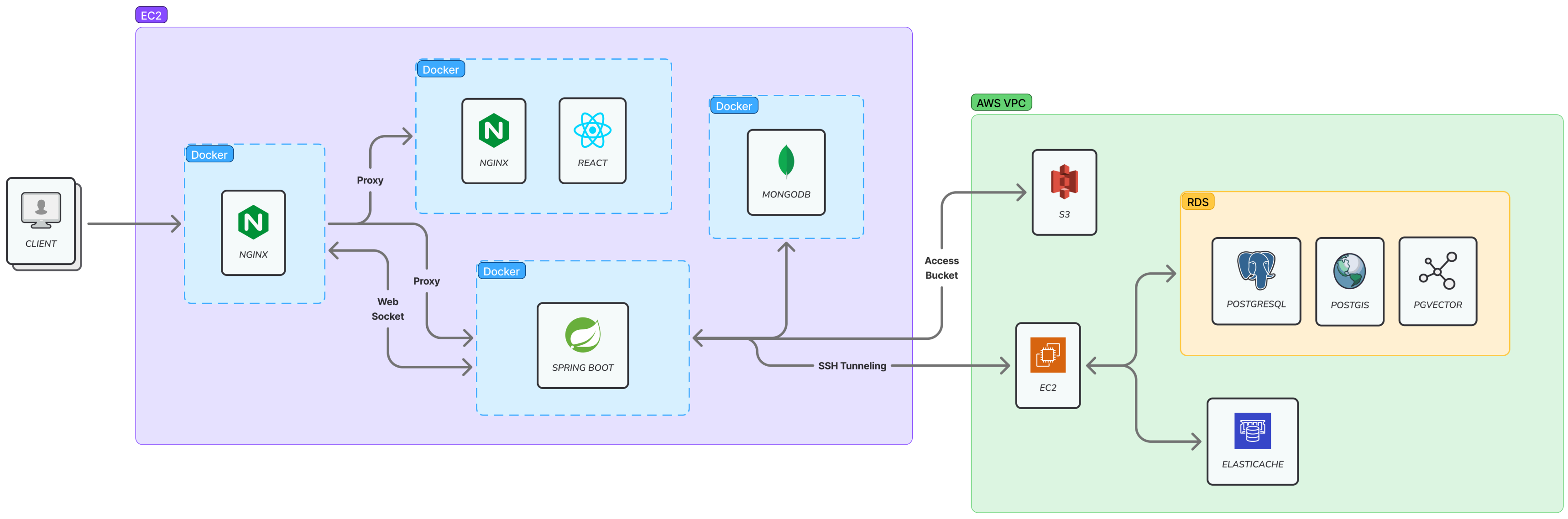
포팅매뉴얼

> **서비스명 : 통학 및 통근 시간 기반의 매물 탐색 부동산 플랫폼, 쌩GO집**
> **팀명 : 희희호호**

1. Architecture



2. Infra

VERSION

- **Domain** : j12a406.p.ssafy.io
- **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
MongoDB	27017
PostgreSQL	5433
Redis	6380

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:
      - nginx-certs:/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 & wait $${!}; done;'"
    networks:
      - app-network

  ######################################################################
  # 3. MongoDB
  ######################################################################
  mongo:
    image: mongo:6.0
    container_name: mongodb
    restart: always
    volumes:
      - mongo-data:/data/db
    ports:
      - "27017:27017"
    networks:
      - app-network

volumes:
  nginx-html:
  nginx-certs:
  mongo-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 j12a406.p.ssafy.io;

    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 j12a406.p.ssafy.io;

    ssl_certificate /etc/nginx/certs/live/j12a406.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/nginx/certs/live/j12a406.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://j12a406.p.ssafy.io:8080/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
    location / {
        proxy_pass http://j12a406.p.ssafy.io:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /api/ws-chat {
        proxy_pass http://j12a406.p.ssafy.io:8080/ws-chat;


        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_read_timeout 3600s;
    }
}
```

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 j12a406.p.ssafy.io \
  --email your@email.com \
  --agree-tos \
  --no-eff-email
```

8. **Nginx 재실행**

```
docker restart nginx-proxy
```

SSH 터널링

1. `SUPERAPP-User-S12P21A406-Key.pem` **준비**
2. **PostgreSQL (AWS RDS) 연결**

```
ssh -f -N -L 5433:rds-postgres-stg.ssafyapp.com:5432 a406hiho@15.164.101.122 -i SUPERAPP-User-S12P21A406-Key.pe
```

3. **Redis (AWS Elasticache) 연결**

```
ssh -f -N -L 6380:redis-stg.ssafyapp.com:6379 a406hiho@15.164.101.122 -i SUPERAPP-User-S12P21A406-Key.pem
```

3. Frontend

VERSION

- **nodeJs** : 22
- **Vite** : 6.2.2
- **React** : 19.0.0
- **Typescript** : 5.8.2
- **React Router** : 7.4.0
- **Tailwind CSS** : 3.4.17
- **Tanstack Query** : 5.69.0
- **Zustand** : 5.0.3
- **Axios** : 1.8.4
- **STOMP JS**: 7.1.0

실행 가이드

1. **Git clone (frontend 브랜치만)**

```
git clone -b frontend --single-branch https://lab.ssafy.com/s12-webmobile1-sub1/S12P11A402.git
```

2. **폴더 이동**

```
cd S12P11A402/Frontend
```

3. `.env` **파일 작성**

```
VITE_APP_BASE_URL=https://j12a406.p.ssafy.io/api
VITE_KAKAO_MAP_KEY=80d830a065cca6f44b1812e2f0679579
VITE_KAKAO_REST_API_KEY=e63eb660333603e35df6c789bf3d2d63
```

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, MongoDB
- **API Documentation** : SpringDoc OpenAPI 2.8.5
- **Web** : Spring Boot Web
- **Testing Frameworks** : JUnit

실행 가이드

1. **Git clone (backend 브랜치만)**

```
git clone -b backend --single-branch https://lab.ssafy.com/s12-webmobile1-sub1/S12P11A402.git
```

2. **폴더 이동**

```
cd S12P11A402/Backend/ssaeng-go-jip
```

3. `.env` **파일 작성**

```
POSTGRESQL_URL=localhost:5433/a406hiho
POSTGRESQL_USERNAME=a406hiho
POSTGRESQL_PASSWORD=a406!Kspa13

REDIS_HOST=localhost

JWT_SECRET_KEY=DT0JQzUySOUxEDxHV8p0UyvfftDW8UUxgrdmp68F82M=

KAKAO_CLIENT_ID=718ce25c291f8df2a0c47fb96b652c80
KAKAO_CLIENT_SECRET=ejfPxNBnrSjfYOcTtY7UbS8i58wGW4bb
KAKAO_REDIRECT_URL=https://j12a406.p.ssafy.io/account/login/kakao

NAVER_CLIENT_ID=6_93CkviDfSFm_GrMYaL
NAVER_CLIENT_SECRET=W446LxWuzI
NAVER_REDIRECT_URL=https://j12a406.p.ssafy.io/account/login/naver

GOOGLE_CLIENT_ID=809999665942-voo0ol6ab29sin9oqh3ipjlugmg3u6jk.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=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
GOOGLE_REDIRECT_URL=https://j12a406.p.ssafy.io/account/login/google

SSAFY_CLIENT_ID=70ffec42-e719-4a38-9f27-f991ef8b1dce
SSAFY_CLIENT_SECRET=211038a5-6b52-4cc3-80e8-46eec8368e47
SSAFY_REDIRECT_URL=https://j12a406.p.ssafy.io/account/login/ssafy

MAIL_USERNAME=ssaenggojip@gmail.com
MAIL_PASSWORD=uzqjgrztualhaafz
```

```
MONGODB_URL=j12a406.p.ssafy.io:27017/ssaeng-go-jip
```

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
- **pgvector** : 0.8.0
- **Redis** : 7.1.0
- **MongoDB**: 6.0

실행 가이드

1. **PostGIS 설치**

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

2. **pgvector 설치**

```
CREATE EXTENSION IF NOT EXISTS vector;
```

3. **MongoDB에 `ssaeng-go-jip.chatRoom.json` 파일 import**
4. **PostgreSQL에 `a406hiho_backup_nodata.dump` 파일 import**