최종 산출물

▼ 포팅메뉴얼

포팅메뉴얼 - 메인서버

개발 환경

OS

- Ubuntu 20.04

이슈 관리

- Jira

Communication

- Mattermost
- Notion

Front-end

- Vue.js : 16.13.0
- bootstrap : 5.3.2
- Pinia : 2.1.7
- pinia-plugin-persistedstate : 3.2.1
- nodejs : 20.11.0
- Vite : 5.0.11

DB

- MySQL : 8.0.36

- Redis : latest

Back-end

- Java : 17.0.10
- Spring Boot : 3.0.10
- spring-boot-starter-data-jpa
- spring-boot-starter-security
- spring-boot-starter-oath2-client
- JWT : 0.12.3
- Pyhon
- Selenium
- pandas

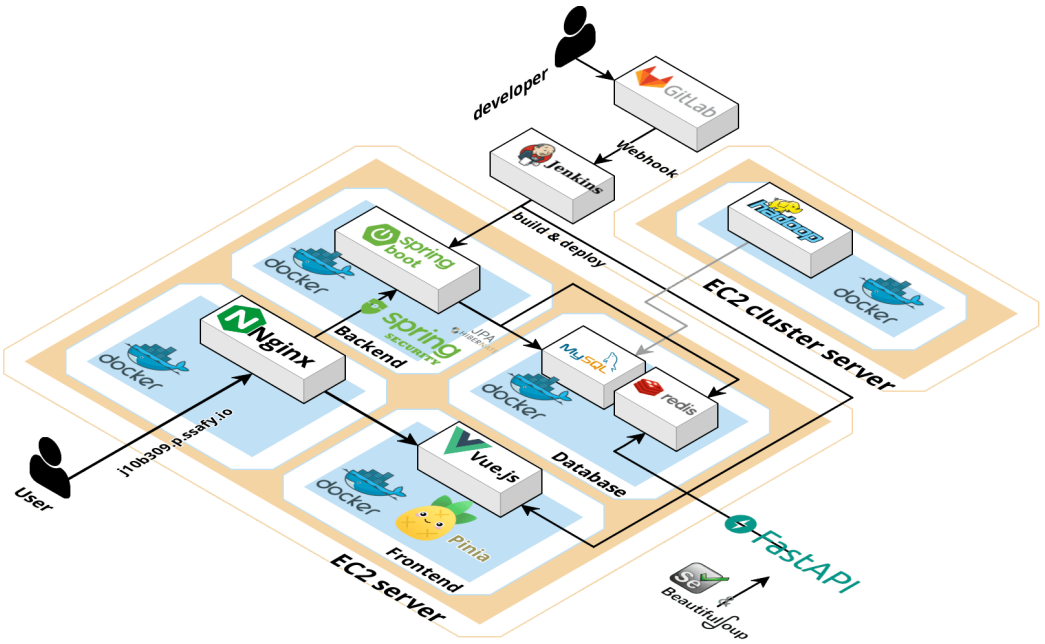
IDE

- IntelliJ
- Visual Studio Code
- Anaconda

Infra - prod server

- Nginx : latest
- Jenkins : lts
- Docker : 26.0.0
- docker-compose : 2.25.0

서비스 아키텍처



EC2 초기 설정

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install build-essential
$ sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

JAVA 설치

```
$ sudo apt install openjdk-17-jdk
```

Docker 설치

```
$ sudo apt update
$ sudo apt install apt-transport-https ca-certificates curl
$ sudo wget -qO- https://get.docker.com/ | sh
```

docker-compose 설치

```
$ sudo curl -L https://github.com/docker/compose/releases/l

# Docker Compose 실행 권한 부여
$ sudo chmod +x /usr/local/bin/docker-compose
```

Nginx, DB, fastAPI (docker-compose.yml)

```
services:
  nginx:
    container_name: nginx
    image: nginx:latest
    restart: always
    volumes:
      - ./config/default.conf:/etc/nginx/conf.d/default.cor
      - /etc/letsencrypt:/etc/letsencrypt
      - /var/log/nginx:/var/log/nginx
    ports:
      - 80:80
      - 443:443
    environment:
      - TZ=Asia/Seoul
  mysql:
    container_name: mysql
    image: mysql:8.0.36
    restart: always
    environment:
      MYSQL_DATABASE: jobis_db
      MYSQL_USER: [USERNAME]
      MYSQL_PASSWORD: [PASSWORD]
      MYSQL_ROOT_PASSWORD: [ROOT_PASSWORD]
    volumes:
      - ./mysql:/var/lib/mysql
      - ./config/mysql-config:/etc/mysql/mysql.conf.
    ports:
      - 3306:3306
```

```
  redis:
    container_name: redis
    image: redis:latest
    restart: always
    volumes:
      - ./data/redis:/data
      - ./config/redis.conf:/usr/local/conf/redis.conf
    ports:
      - 6379:6379
    command: redis-server /usr/local/conf/redis.conf

    fastapi-app:
    container_name: fastapi
    image: fastapi
    restart: always
    ports:
      - 8000:8000

networks:
  default:
    name: jobis-network
    driver: bridge
```

Nginx (default.conf)

```
# HTTP 서버 설정 (80 포트)
server {
    listen 80;
    server_name j10b309.p.ssafy.io;

    # 모든 HTTP 요청을 HTTPS로 리다이렉트
    location / {
        return 301 https://$server_name$request_uri;
    }
}

# HTTPS 서버 설정 (443 포트)
```

```
server {
    listen 443 ssl;
    server_name j10b309.p.ssafy.io;

    # SSL 인증서 경로 설정
    ssl_certificate /etc/letsencrypt/live/j10b309.p.ssafy.i
    ssl_certificate_key /etc/letsencrypt/live/j10b309.p.ssa

    location /api {
        proxy_pass http://j10b309.p.ssafy.io:8081; # Spring
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwa
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location / {
        proxy_pass http://j10b309.p.ssafy.io:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwa
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

fastAPI (.env)

```
API_KEY=[OpenAI_API_KEY]
BASE_URL=https://search.naver.com/search.naver
```

Jenkins(Dockerfile)

```
FROM jenkins/jenkins:lts
USER root
#RUN apt-get update &&\
#    apt-get upgrade -y &&\
#    apt-get install -y openssh-client
```

```
RUN apt-get update && \
    apt-get -y install apt-transport-https \
      ca-certificates \
      curl \
      gnupg2 \
      software-properties-common && \
    curl -fsSL https://download.docker.com/linux/$(. /etc/c
    add-apt-repository \
      "deb [arch=amd64] https://download.docker.com/linux/$
      $(lsb_release -cs) \
      stable" && \
   apt-get update && \
   apt-get -y install docker-ce docker-ce-cli

# docker-compose 버전 2.25.0 설치
RUN curl -L "https://github.com/docker/compose/releases/dow
    chmod +x /usr/local/bin/docker-compose

RUN groupadd -f docker
RUN usermod -aG docker jenkins
```

Jenkins (docker-compose.yml)

```
services:
  jenkins:
    container_name: jenkins-compose
    build:
      context: .
      dockerfile: Dockerfile
    user: root
    ports:
    - 8080:8080
    volumes:
      - /home/ubuntu/develop/jenkins/jenkins-data:/var/jenk
      - /var/run/docker.sock:/var/run/docker.sock
      - /home/ubuntu/develop/config/environment/.env:/var/j
```

backend (Dockerfile)

```
FROM openjdk:17-jdk
CMD ["./gradlew", "clean", "build"]
ARG JAR_FILE=build/libs/*.jar
# 애플리케이션의 JAR 파일을 컨테이너에 app.jar라는 이름으로 복사
# 실제 build된 jar 파일 경로와 build.gradle을 참고하여 파일 이름을
COPY ${JAR_FILE} app.jar
EXPOSE 8081
ENTRYPOINT ["java","-jar","/app.jar"]
```

frontend (Dokcerfile)

```
# Node.js 공식 이미지를 기반으로 설정
FROM node:20.11.1

# 작업 디렉토리 설정
WORKDIR /app

# 앱 의존성 설치
COPY package*.json ./
RUN npm install

# 앱 소스 복사
COPY . .

# 앱 빌드
RUN npm run build

# 3000 포트 노출
EXPOSE 3000

CMD ["npm", "run", "dev"]
```

Spring boot, Vue.js (docker-compose.yml)

```yaml
services:
  spring-boot-app:
    container_name: backend
    image: jobis-backend  # Docker에서 빌드한 Spring Boot 애플
    restart: unless-stopped
    ports:
      - 8081:8081
    environment:
      JWT_SECRET_KEY: ${JWT_SECRET_KEY}
      JWT_ACCESS_EXPIRED: ${JWT_ACCESS_EXPIRED}
      JWT_REFRESH_EXPIRED: ${JWT_REFRESH_EXPIRED}
      DATABASE_URL: ${DATABASE_URL}
      DATABASE_USERNAME: ${DATABASE_USERNAME}
      DATABASE_PASSWORD: ${DATABASE_PASSWORD}
      REDIS_HOST: ${REDIS_HOST}
      REDIS_PORT: ${REDIS_PORT}
      CORS_ALLOWED_ORIGINS: ${CORS_ALLOWED_ORIGINS}
      GOOGLE_CLIENT_ID: ${GOOGLE_CLIENT_ID}
      GOOGLE_CLIENT_SECRET: ${GOOGLE_CLIENT_SECRET}
      NAVER_CLIENT_ID: ${NAVER_CLIENT_ID}
      NAVER_CLIENT_SECRET: ${NAVER_CLIENT_SECRET}
      KAKAO_CLIENT_ID: ${KAKAO_CLIENT_ID}
      KAKAO_CLIENT_SECRET: ${KAKAO_CLIENT_SECRET}

  vue-app:
    container_name: frontend
    image: jobis-front
    restart: unless-stopped
    ports:
      - 3000:3000
    environment:
      VITE_APP_API_URL: ${VITE_APP_API_URL}

networks:
  default:
    external:
      name: jobis-network
```

Spring boot, Vue.js (.env)

```
#jwt
JWT_SECRET_KEY=[JWT_SECRET_KEY]
JWT_ACCESS_EXPIRED=1800000
JWT_REFRESH_EXPIRED=604800000

#mysql
DATABASE_URL=j10b309.p.ssafy.io/jobis_db
DATABASE_USERNAME=[USERNAME]
DATABASE_PASSWORD=[PASSWORD]

#redis
REDIS_HOST=redis
REDIS_PORT=6379

#cors
CORS_ALLOWED_ORIGINS=http://localhost:3000, https://j10b309

# aouth
GOOGLE_CLIENT_ID=[GOOGLE_CLIENT_ID]
GOOGLE_CLIENT_SECRET=[GOOGLE_CLIENT_SECRET]
NAVER_CLIENT_ID=[NAVER_CLIENT_ID]
NAVER_CLIENT_SECRET=[NAVER_CLIENT_SECRET]
KAKAO_CLIENT_ID=[KAKAO_CLIENT_ID]
KAKAO_CLIENT_SECRET=[KAKAO_CLIENT_SECRET]

#front
VITE_APP_API_URL = https://j10b309.p.ssafy.io/api
```

포팅메뉴얼 - 데이터서버

개발환경

- fastapi : 0.78.0
- python-dotenv : 0.20.0
- requests : 2.27.1

- beautifulsoup4 : 4.10.0
- uvicorn : 0.17.6

OS

- Ubuntu 20.04

이슈 관리

- Jira

Communication

- Mattermost
- Notion

Infra

- Docker(26.0.0)
- docker-compose(2.25.0)

Infra - Hadoop cluster server

- centos
- hadoop : 3.3.6
- java : 1.8.0_312