

최종 산출물

배포시 특이사항

- Jenkins에서 pipeline을 작성하여, GitLab의 특정 브랜치에 새로운 내용이 merge될 때마다 자동으로 GitLab의 코드를 clone하고 자동 배포하도록 하였습니다.
- 백엔드에서 테스트 코드 작성을 통해 배포 시, 비정상적으로 동작하는 프로그램이 동작하지 않도록 하였습니다.
- Nginx를 사용하여, 클라이언트로부터 온 요청 중 /api로 시작하면 백엔드로, /로 오면 프론트엔드로 도착하도록 했습니다.
- dockerfile을 작성하여, GitLab에서 받아온 코드를 jar 파일로 만든 뒤, 도커 허브에 이미지로 등록했습니다.
- 도커 및 도커 허브에 등록된 이미지를 가져옴으로서 어떤 환경에서도 동일하게 동작되도록 하였습니다.
- 백엔드와 프론트엔드 각각 블루 그린 배포를 통해 업데이트된 애플리케이션을 서버에 배포해도, 중간에 서비스가 끊기지 않아 가용성을 높힐 수 있었습니다.

백엔드

1. JVM

- **JVM 버전:** Java 17
- **설정:** Gradle에서 Java 17을 사용하도록 설정 (`toolchain` 에서 지정)

2. 웹 서버 및 WAS

- **웹 서버:** 내장형 Tomcat (Spring Boot의 기본 설정)
- **WAS:** 내장형 서버 사용 (Spring Boot가 내장 Tomcat을 제공하며, 별도의 외부 WAS 설정은 없음)

3. Gradle 플러그인

- **플러그인:**
 - `java`: 기본 Java 플러그인
 - `org.springframework.boot` - Spring Boot 플러그인
 - 버전: 3.3.2
 - `io.spring.dependency-management` - Spring Dependency Management 플러그인
 - 버전: 1.1.6

4. 의존성

- **Spring Boot Starter**
 - `spring-boot-starter-data-jpa`
 - `spring-boot-starter-security`
 - `spring-boot-starter-web`
 - `spring-boot-starter-oauth2-client`
 - `spring-boot-starter-mail`
 - `spring-boot-starter-validation`
- **Lombok**
 - `lombok` (compileOnly 및 annotationProcessor로 설정)
- **테스트**
 - `spring-boot-starter-test`

- `spring-security-test`
- `h2` (테스트 데이터베이스)
- `junit-platform-launcher` (테스트 런처)
- **JWT 라이브러리**
 - `java-jwt` (버전 4.4.0)
- **SpringDoc**
 - `springdoc-openapi-starter-webmvc-ui` (버전 2.2.0)
- **MySQL 커넥터**
 - `mysql-connector-j` (runtimeOnly 설정)

5. IDE 버전

- **IDE:** 명시된 IDE 버전은 제공되지 않았으므로, Gradle 및 Spring Boot의 최신 버전과 호환되는 IDE를 사용하는 것이 좋습니다 (예: IntelliJ IDEA, Eclipse, Visual Studio Code 등).

6. 기타 설정

- **테스트 설정:**
 - JUnit Platform 사용 (`tasks.named('test')` 에서 `useJUnitPlatform()` 으로 설정)

7. DB 접속 정보

```

DATABASE_URL: jdbc:mysql://localhost:3306/closetoyou?serverTimezone=UTC
DATABASE_DRIVER_CLASS_NAME: com.mysql.cj.jdbc.Driver
DATABASE_ID: root
DATABASE_PW: 1234

```

프론트엔드

1. .env 파일

```

VITE_CLIENT_URL="http://localhost:3000/"
VITE_REST_API_URL="http://localhost:8080/"

VITE_KAKAO_LOGIN_URL="http://localhost:8080/api/oauth2/authorization/kakao"
VITE_GOOGLE_LOGIN_URL="http://localhost:8080/api/oauth2/authorization/google"

; VITE_CLIENT_URL="https://i11b201.p.ssafy.io/"
; VITE_REST_API_URL="https://i11b201.p.ssafy.io/"

; VITE_KAKAO_LOGIN_URL="https://i11b201.p.ssafy.io/api/oauth2/authorization/kakao"
; VITE_GOOGLE_LOGIN_URL="https://i11b201.p.ssafy.io/api/oauth2/authorization/google"

```

2. node version

`v20.15.0`

3. package.json

```

{
  "name": "s11p12b201",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",
    "format": "prettier --cache --write .",
    "preview": "vite preview",
    "start": "vite preview --host"
  },
  "dependencies": {
    "@emotion/css": "^11.13.0",
    "@emotion/react": "^11.13.0",
    "@emotion/styled": "^11.13.0",
    "axios": "^1.7.3",
    "js-cookie": "^3.0.5",
    "react": "^18.3.1",
    "react-dom": "^18.3.1",
    "react-router-dom": "^6.25.1",
    "react-swipeable": "^7.0.1",
    "zustand": "^4.5.4"
  },
  "devDependencies": {
    "@swc/cli": "^0.4.0",
    "@swc/core": "^1.7.0",
    "@types/react": "^18.3.1",
    "@types/react-dom": "^18.3.0",
    "@vite-pwa/assets-generator": "^0.2.4",
    "@vitejs/plugin-react": "^4.2.1",
    "eslint": "^8.57.0",
    "eslint-config-prettier": "^9.1.0",
    "eslint-plugin-react": "^7.35.0",
    "eslint-plugin-react-hooks": "^4.6.2",
    "eslint-plugin-react-refresh": "^0.4.6",
    "prettier": "^3.3.3",
    "vite": "^5.2.10",
    "vite-plugin-pwa": "^0.20.0",
    "vite-plugin-svgr": "^4.2.0",
    "workbox-core": "^7.1.0",
    "workbox-precaching": "^7.1.0",
    "workbox-routing": "^7.1.0",
    "workbox-strategies": "^7.1.0"
  },
  "overrides": {
    "sharp": "0.32.6",
    "sharp-ico": "0.1.5"
  }
}

```