

# Gitlab 소스 클론 이후 빌드/배포 시 참고사항

## 1. 개발 환경

- 사용한 JVM
  - BellSoft LibericaJDK-17
- 웹서버/WAS
  - Spring boot 3.4.5
- IDE
  - IntelliJ IDEA 2024.3.1.1
  - Cursor IDEA

## 2. 빌드 시 사용되는 환경변수

### ■ .env (/app/.env)

```
■ NODE_ENV=development
■ CRAWLER_API_BASE_URL=배포한서버주소/v3/mcp/servers
■ GUI_BE_API_BASE_URL=http://localhost:8082/api/v1
■ SERVER_NAME=MCP Fallback Server
■ SERVER_VERSION=0.1.0
```

### ■ secret.yml (backend/src/main/resources/secret.yml)

```
■ # 실제 사용 시 이 파일명을 secret.yml로 변경하고 실제 값으로 채워주세요.
■ # 주석(#)으로 시작하는 라인은 실제 설정에 영향을 주지 않습니다.
■
■ # MongoDB (자체 호스팅 또는 클라우드 인스턴스)
■ mongodb:
■     host: "your-mongodb-host.example.com" # 예: k12a201.p.ssafy.io 또는 localhost
■     port: 27017 # MongoDB 기본 포트
```

```

■ database: "your_mongodb_database_name" # 예: mcp_v1
■ username: "your_mongodb_user" # 예: a201
■ password: "your_mongodb_password" # 예:
  "replace_with_actual_password"
■ # atlas: # MongoDB Atlas 사용 시 추가 설정 예시 (현재
  프로젝트에서는 위 기본 설정을 사용하는 것으로 보임)
■ # username: "your_atlas_user"
■ # password: "your_atlas_password"
■ # cluster: "your_atlas_cluster_name.mongodb.net" # 예:
  scp201.lp4wshh
■ # app-name: "your_atlas_app_name" # 예: scp201 (연결 문자열
  옵션에 따라 다를 수 있음)
■ # database: "your_atlas_database_name" # 예: mcplink
■
■ # MySQL
■ mysql:
■ host: "your-mysql-host.example.com" # 예: mcplink.co.kr 또는
  localhost
■ port: 3306 # MySQL 기본 포트
■ user: "your_mysql_user" # 예: a201
■ password: "your_mysql_password" # 예:
  "replace_with_actual_password"
■ database: "your_mysql_database_name" # 예: mcplink
■
■ # 애플리케이션 특정 설정
■ app:
■ analysis:
■ temp-dir: "/app/analysis_temp" # 분석용 임시 디렉토리 경로
  (리눅스 서버 기준 예시)
■ # 로컬 개발 시 예: "C:/dev/mcplink_temp_analysis"
■ osv-scanner-cmd: "osv-scanner" # OSV-Scanner 실행 명령어
■
■ # SSAFY OAuth 2.0 설정
■ ssafy:
■ oauth:

```

```

    client-id: "your_ssafy_oauth_client_id" # 예:
    "64da588a-f264-4a90-924c-7cb326bf9d3a"

    client-secret: "your_ssafy_oauth_client_secret" # 예:
    "52c55ce6-cb6c-48d4-b17f-433e4178bd9c"

    redirect-uri:
    "https_or_http://your_app_domain/api/v1/auth/ssafy/callback" #
    애플리케이션의 콜백 URL

    # 예: "https://mcplink.co.kr/api/v1/auth/ssafy/callback"

    # 로컬 테스트 시:
    "http://localhost:8080/api/v1/auth/ssafy/callback"

    authorization-uri:
    "https://project.ssafy.com/oauth/sso-check" # SSAFY 인증
    엔드포인트

    token-uri: "https://project.ssafy.com/ssafy/oauth2/token" #
    SSAFY 토큰 발급 엔드포인트

    user-info-uri:
    "https://project.ssafy.com/ssafy/resources/userInfo" # SSAFY
    사용자 정보 엔드포인트

    scope: "email name" # 요청할 사용자 정보 범위

# JWT (JSON Web Token) 설정
jwt:

    secret:
    "your_very_long_and_secure_jwt_secret_key_at_least_256_bits_pr
    eferably_512_bits_for_hs512"

    access-expiration: "3600000" # Access Token 만료 시간

    refresh-expiration: "604800000"

    secure-path: # 인증이 필요한 경로 목록

    - "/v1/posts"
    - "/v1/posts/**"
    - "/v1/users/info"
    # - "/another/secure/path"

    oauth-ssafy-path: # 인증 필터를 통과해야 하지만 JWT 인증은 요구하지
    않는 경로 (OAuth 콜백, Swagger)

    - "/auth/ssafy/login"
    - "/auth/ssafy/callback"
    - "/swagger-ui/**" # Swagger UI 경로
    - "/v3/api-docs/**" # OpenAPI 문서 경로

```

```

■ # - "/public/path/example"
■
■ # GitHub API 설정
■ github:
■   token: "your_github_personal_access_token_or_oauth_token" #
  GitHub API 접근을 위한 토큰
■
■ # Google Gemini API 설정
■ gemini:
■   api-key: "your_google_gemini_api_key" # Google Gemini API 키

```

- /exec 내용 속 샘플 참고

### 3. 배포 시 특이사항

- 배포 환경
  - 운영 환경에서는 Docker로 생성한 Jenkins에서 빌드 파일을 Docker Hub에 Push 하고 EC2 서버에서 pull 받고 실행시키는 방식으로 자동 배포
- 로컬 환경
  - 백엔드 어플리케이션 실행
  - 타우리 앱 빌드
    - ◆ cargo tauri build