**GitLab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서**

**1.사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전(IDE버전 포함) 기재**  
사용한 JVM

openjdk version "17.0.13" 2024-10-15

OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu120.04)

OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu120.04, mixed mode, sharing)

웹서버  
mainServer   
역할: 로그인, 캐릭터 정보, 뉴스, 로그인 등 기본 api   
이미지: main-server   
컨테이너: main-server-container  
포트: 8080

crawlingServer  
역할: 경기시작 스케줄링으로 시작하여 KBO 문자 중계 사이트의 데이터를 DOM변화 감지 후 경기내용 파싱 후 Kafka로 보냄  
이미지: crawling-crawling-server  
컨테이너: crawling-server-container  
포트: 5000

sseServer  
역할: Kafka의 각 경기의 실시간 데이터를 SSE 프로토콜을 통해 적합한 유저에게 전달  
이미지: sse-server  
컨테이너: sse-server-container  
포트: 8081

Nginx  
역할 : 리버시 프록시로 클라이언트와 연결되 sseServer와 mainServer가 사용

포트: 80

WAS제품

**Electron 애플리케이션**역할 : React, html 등 웹 기술을 활용하여 데스크탑 앱으로 빌드하기 위한 프레임워크  
기술 스택: React 18.3.1, node.js 20.18.0, Electron 32.2.1, TypeScript 5.5.3  
IDE: VSCode 1.95.1

mainServer  
기술 스택: Spring Boot 3.3.5  
IDE: IntelliJ IDEA 2024.2.0.2 (Ultimate Edition)

crawlingServer  
기술 스택: Python 3.10.11  
IDE: PyCham 2024.2

sseServer  
기술 스택: Spring Boot 3.3.5  
IDE: IntelliJ IDEA 2024.2.0.2 (Ultimate Edition)

**2.빌드시 사용되는 환경 변수 등의 내용 상세 기재**

mainServer   
application.yml – GitLab에 존재  
application-dev.yml - GitLab에 존재  
.env   
SPRING\_KAKAO\_AUTH\_CLIENT=3220f8022636c44b43a03127f68aea11

SPRING\_KAKAO\_AUTH\_REDIRECT\_URL=http://localhost:8080/bf/auth/login/kakao

SPRING\_KAKAO\_AUTH\_USER\_INFO\_URL=https://kapi.kakao.com/v2/user/me

JWT\_ACCESS\_EXPIRATION=10080

JWT\_SECRET\_KEY=baseballfriendbaseballfriend123456789

MYSQL\_PASSWORD=songsongsong

MYSQL\_SERVER=k11a505.p.ssafy.io:3306

MYSQL\_USERNAME=ssafy

NAVER\_CLIENT\_ID=b9JPJKumLkptec1br6yb

NAVER\_CLIENT\_SECRET=tFa1roAZla

NAVER\_NEWS\_SEARCH\_API=https://openapi.naver.com/v1/search/news.json?display=100&sort=date&query=

FLASK\_REQUEST\_URL=http://k11a505.p.ssafy.io:5000/flask/bf

OPEN\_AI\_API\_KEY=sk-proj-3bWmLJ-dcOQg1Qco0UQyBSS0LTdtu4HqIBNkUp8dG66NIsvbP9m0Jd\_2lRK24Z5mqKT4k9x7vFT3BlbkFJR\_sQ1hui5BgUzYNTwLxaIu8KPIE3FbY7vwfszpWzfHeLtsE-DLLyDzxjLBIwDUn674pFGBoc0A

OPEN\_AI\_IMAGE\_API\_KEY=sk-proj-hRKUPttXWZ788YMRfvwz7xZ9\_oyNKwnqOW-3T6G9I9FqeP8jAKvZIOzYUIRdgulp7UQTSIw7ggT3BlbkFJTIuAW-z2R58ZZj5vPjxzPoLAiWswZSLoIfktVee8CyuXVJ5PPE5QduZPeCIN9M92xQ19ODh9kA

OPEN\_AI\_TEXT\_GENERATE\_API=https://api.openai.com/v1/chat/completions

OPEN\_AI\_IMAGE\_GENERATE\_API=https://api.openai.com/v1/images/generations

SPRING\_MAIL\_USERNAME=baseballfrienda505@gmail.com

SPRING\_MAIL\_PASSWORD=dcnhtkwbcvgsdffk

SPRING\_DATA\_REDIS\_HOST=k11a505.p.ssafy.io

SPRING\_DATA\_REDIS\_PORT=6379

SPRING\_DATA\_REDIS\_DURATION=10

s3\_access\_key=AKIAXAJL2LXMUHZFPUWL

s3\_secret\_key=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

s3\_bucket\_name=best-friend-bucket

s3\_region=ap-southeast-2  
  
sseServer   
application.yml – GitLab에 존재  
application-dev.yml - GitLab에 존재  
.env  
BOOTSTRAP\_SERVER=k11a505.p.ssafy.io:9092

REDIS\_HOST=k11a505.p.ssafy.io

REDIS\_PORT=6379

JWT\_SECRET\_KEY=baseballfriendbaseballfriend123456789

MYSQL\_PASSWORD=songsongsong

MYSQL\_SERVER=k11a505.p.ssafy.io:3306

MYSQL\_USERNAME=ssafy  
  
crawlingServer  
.env  
KBOID=09hyen28

KBOPW=teamjhw505

MATCH\_LIVE\_URL\_PREFIX=https://www.koreabaseball.com

LIVE\_URL=http://k11a505.p.ssafy.io:3000

RESULT\_URL=http://k11a505.p.ssafy.io:8080/bf/match/result

Elethron  
.env  
VITE\_OPENAI\_API\_KEY='sk-proj-3bWmLJ-dcOQg1Qco0UQyBSS0LTdtu4HqIBNkUp8dG66NIsvbP9m0Jd\_2lRK24Z5mqKT4k9x7vFT3BlbkFJR\_sQ1hui5BgUzYNTwLxaIu8KPIE3FbY7vwfszpWzfHeLtsE-DLLyDzxjLBIwDUn674pFGBoc0A'

**3.배포시 특이사항 기재**

Nginx의 리버시 프록시   
mainServer <http://k11a505.p.ssafy.io:8080/bf> -> http://k11a505.p.ssafy.io/bf  
sseServer <http://k11a505.p.ssafy.io:8081/bf/sse> -> http://k11a505.p.ssafy.io/bf/sse   
젠킨스로 자동 CI/CD 배포

**4.DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록**

DB 접속 정보  
DB 서버 : k11a505.p.ssafy.io:3306  
DB 사용자 : ssafy  
DB 비밀번호 : songsongsong  
DB 이름 : best\_friend  
  
ERD에 활용되는 주요 계정  
member : 회원 정보  
team : 구단명과 구단의 위치 정보  
team\_rank : 팀의 현재 순위와 세부 정보  
match\_info : 과거 현재 미래에 대한 경기일정에 대한 정보  
match\_prediction : 어느 팀이 승리를 하는지 예측하는 정보  
news : 네이버 뉴스를 AI를 활용하여 요약 후 저장  
character\_info : 이 어플에 존재하는 캐릭터들의 가격과 일련번호 등의 정보  
member\_character : 회원이 보유하는 캐릭터들의 정보  
item : 캐릭터에 착용이 가능한 아이템의 가격과 일련번호 등의 정보  
member\_item : 회원이 보유한 아이템의 정보  
background : 캐릭터의 배경에 대한 가격과 일련번호 정보  
member\_background : 회원이 보유한 배경의 정보  
  
프로퍼티가 정의된 파일 목록

mainServer  
- .env 위에 존재  
- application.yml – GitLab에 존재  
- application-dev.yml - GitLab에 존재  
- Dockerfile – ec2 cat /home/Ubuntu/mainServer/Dockerfile - 존재  
- docker-compose.yml – ec2 cat /home/Ubuntu/mainServer/docker-compose.yml - 존재  
- Jenkins pipeline - <http://k11a505.p.ssafy.io:9090/> 접속 ID : ssafy PW: ssafyssafy 접속 후 mainServer에 pipeline 존재

sseServer  
- .env 위에 존재  
- application.yml – GitLab에 존재  
- application-dev.yml - GitLab에 존재  
- Dockerfile – ec2 cat /home/Ubuntu/sseServer/Dockerfile - 존재  
- docker-compose.yml – ec2 cat /home/Ubuntu/sseServer/docker-compose.yml - 존재  
- Jenkins pipeline - <http://k11a505.p.ssafy.io:9090/> 접속 ID : ssafy PW: ssafyssafy 접속 후 sseServer에 pipeline 존재

crawlingServer  
- .env 위에 존재   
- Dockerfile – ec2 cat /home/Ubuntu/crawlingServer/Dockerfile - 존재  
- docker-compose.yml – ec2 cat /home/Ubuntu/crawlingServer/docker-compose.yml - 존재  
- Jenkins pipeline - <http://k11a505.p.ssafy.io:9090/> 접속 ID : ssafy PW: ssafyssafy 접속 후 crawlingServer에 pipeline 존재

electhron  
-.env 위에 존재  
- vite.config.ts GitLab에 존재  
- package.json GitLab에 존재