## 개발 환경

* Back-end
  + Java 17 ([Azul Zulu](https://www.azul.com/downloads/?version=java-17-lts&package=jdk#zulu))
  + Framework - [Spriong Boot 3.1.7](https://start.spring.io/#!type=gradle-project&language=java&platformVersion=3.1.7&packaging=jar&jvmVersion=17&groupId=com.ssafy&artifactId=develop&name=develop&description=Demo%20project%20for%20BrushBuddy&packageName=com.ssafy.develop&dependencies=web,session,devtools,lombok,mysql,data-mongodb,data-jpa,oauth2-client,data-redis)
  + Gradle

Dependencies

* + - Spring Web
    - Spring Session
    - Spring Boot DevTools
    - Lombok
    - MySQL Driver
    - Spring Data JPA
    - OAuth2 Client \*
    - Spring Data Redis (Access+Driver) \*
    - Querydsl
    - Websocket
    - Spring docs Swagger 2.2.0
* Front-end
  + Framework - Vue.js

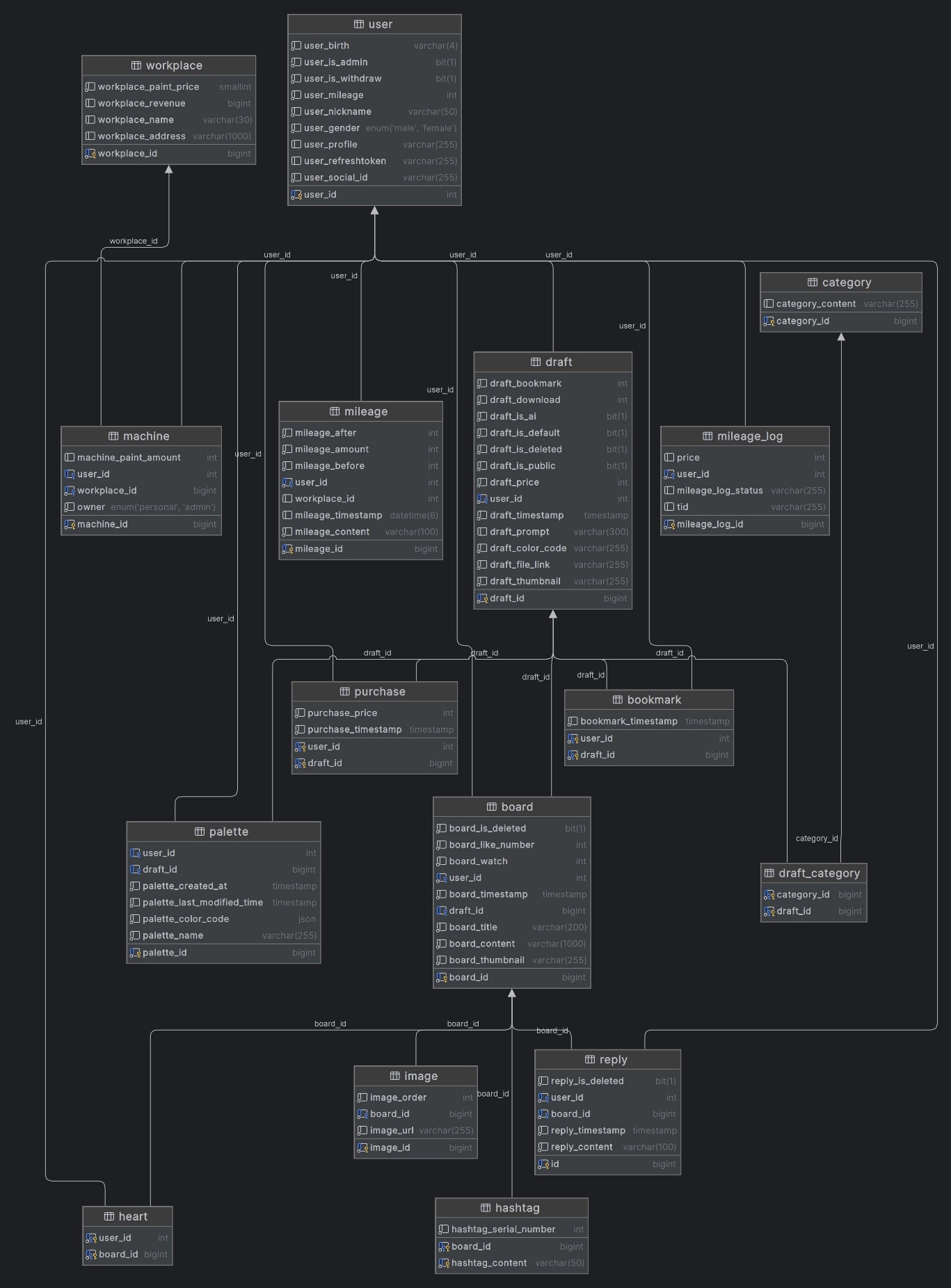
nodejs 20.11.0 LTS

* AI
  + Python 3.10
  + Framwork – fastapi
* RDBMS
  + [MySQL - 8.0.36](https://dev.mysql.com/downloads/installer/)
* Server
  + AWS EC2
  + Tomcat [9.0.85](https://tomcat.apache.org/download-90.cgi)
* HW
  + Framework – Fastapi 5.0.1
  + Python 3.10
  + Mosquitto 2.0.18
  + Raspberry pi 4(bookwarm)
  + Arduino UNO
* References
  + [EasyPipo](https://github.com/AutoPipo/EasyPipo)
  + [kakaosync](https://developers.kakao.com/docs/latest/ko/kakaosync/common) - [앱 정보](https://developers.kakao.com/console/app/1022265)
  + [kakaopay](https://developers.kakao.com/docs/latest/ko/kakaopay/common)

## Tools \*

IntelliJ 2023.3.4, STS, VSCode, Git, GitLab, Jira, Swagger, JWT, Pycharm

[Notion](https://www.notion.so/A205-4c8e57e940f5429d8d81248c7db9a3b7?pvs=21), [Figma](https://www.figma.com/files/project/183254133/brushbuddy?fuid=1324584233444842657), [draw.io](https://app.diagrams.net/#G1dmoe6tuOntTlf1Zn1_2CUiZdAjtjzYZJ), [ERDCloud](https://www.erdcloud.com/d/j2iEGkzmS39Ffz6eh), [Mattermost](https://meeting.ssafy.com/s10p10a2/channels/a205), Discord, Google Drive



* **MYSQL 계정**

**Username : root**

**Password : Ssafya205!**

**2. 배포 시 환경 변수**

**.env**

**================================================**

**MYSQL\_URL=i10a205.p.ssafy.io**

**MYSQL\_PORT=3306**

**MYSQL\_DB=brushBuddy**

**MYSQL\_USERNAME=root**

**MYSQL\_PASSWORD=Ssafya205!**

**S3\_ACCESS\_KEY=AKIA3RPMVVCHMN3JBTMD**

**S3\_SECRET\_KEY=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**

**S3\_BUCKET=brush-buddy-prod**

**KAKAO\_ADMIN\_KEY=a7d115c350be34144764ea1ecbbcac51**

**KAKAO\_CLIENT\_ID=2954ae863e5b58338ac9773723ea466d**

**KAKAO\_CLIENT\_SECRET=csM1qC31YmUpHg2Rt74s3EOOBKqdzHWI**

**KAKAO\_REDIRECT\_URL=https://brush-buddy.duckdns.org/login**

**JWT\_SECRET\_KEY=QDHghlnHjwAzvVRRbh6NQm6dTfoaa3**

**OPEN\_AI\_KEY=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**

**REDIS\_URL=i10a205.p.ssafy.io**

**REDIS\_PORT=6379**

**==================================================**

**3. 배포시 특이사항**

젠킨스를 통해 도커를 자동배포하도록 구현하였기 때문에

Gitlab 소스 클론 이후 추가 빌드 과정은 불 필요합니다.

(관련 설정 배포 설정사항은 하단에 도커 컴포즈 파일을 참조).

* 참조
* 내부 빌드 명령

Frontend – npm install 후 npm run build 수행

Backend – gradlew clean build

* + Docker compose 파일

========================================

version: "3"

services:

jenkins:

image: jenkins/jenkins:lts

container\_name: jenkins

user: root

volumes:

- /home/ubuntu/jenkins-data:/var/jenkins\_home

- /var/run/docker.sock:/var/run/docker.sock

ports:

- 8080:8080

mysql:

image: mysql:8.0.36

container\_name: mysql

environment:

TZ: Asia/Seoul

MYSQL\_ROOT\_PASSWORD: ${MYSQL\_PASSWORD}

ports:

- ${MYSQL\_PORT}:3306

volumes:

- ./db/mysql/data:/var/lib/mysql

- ./db/mysql/init:/docker-entrypoint-initdb.d

spring:

image: fangdol888onssafy/brush-buddy-spring:latest

container\_name: brush-buddy-spring

environment:

MYSQL\_URL: ${MYSQL\_URL}

MYSQL\_PORT: ${MYSQL\_PORT}

MYSQL\_DB: ${MYSQL\_DB}

MYSQL\_USERNAME: ${MYSQL\_USERNAME}

MYSQL\_PASSWORD: ${MYSQL\_PASSWORD}

S3\_ACCESS\_KEY: ${S3\_ACCESS\_KEY}

S3\_SECRET\_KEY: ${S3\_SECRET\_KEY}

S3\_BUCKET: ${S3\_BUCKET}

KAKAO\_ADMIN\_KEY: ${KAKAO\_ADMIN\_KEY}

KAKAO\_CLIENT\_ID: ${KAKAO\_CLIENT\_ID}

KAKAO\_CLIENT\_SECRET: ${KAKAO\_CLIENT\_SECRET}

KAKAO\_REDIRECT\_URL: ${KAKAO\_REDIRECT\_URL}

JWT\_SECRET\_KEY: ${JWT\_SECRET\_KEY}

ports:

- 3000:8080

vue:

image: fangdol888onssafy/brush-buddy-vue:latest

container\_name: brush-buddy-vue

ports:

- 3001:80

volumes:

- ./nginx/nginx.conf:/etc/nginx/nginx.conf

- ./nginx/default.conf:/etc/nginx/conf.d/default.conf

ai:

image: summa1234/brushbuddy:latest

container\_name: brush-buddy-ai

environment:

OPEN\_AI\_KEY: ${OPEN\_AI\_KEY}

S3\_ACCESS\_KEY: ${S3\_ACCESS\_KEY}

S3\_SECRET\_KEY: ${S3\_SECRET\_KEY}

ports:

- 3002:8000

deploy:

resources:

limits:

cpus: "0.50"

memory: "1024M"

reservations:

cpus: "0.25"

memory: "256M"

iot:

image: fangdol888onssafy/brush-buddy-iot:latest

container\_name: brush-buddy-iot

ports:

- 3004:3004

========================================