**[SSAFY 12기 A307팀]**

1. **시스템 환경 설정(사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전(IDE버전 포함) 기재)**
   1. **Backend(API)**
      1. JDK : Oracle OpenJDK 24.0.1
      2. WAS : Apache Tomcat (Spring Boot 내장)
      3. Domain : k12a307.p.ssafy.io
      4. Image
         1. openjdk:24-jdk-slim
         2. port : 8080 -> 8080
         3. context path : /
      5. Development Tool
         1. IDE : IntelliJ IDEA 2024.3.3
         2. Build Tool : Gradle 8.14
      6. Framework : Spring Boot 3.4.5
   2. **Backend(Chatting)**
      1. JDK : Oracle OpnJDK 24.0.1
      2. WAS : Apache Tomcat (Spring Boot 내장)
      3. Domain :
      4. Image
         1. openjdk:24-jdk-slim
         2. port :
         3. context path : /
      5. Development Tool
         1. IDE : IntelliJ IDEA 2024.3.3
         2. Build Tool : Gradle 8.14
      6. Framework : Spring Boot 3.4.5
   3. **Frontend**
      1. JS Runtime : node 22.13.0
      2. Image
         1. node:22-alpine
         2. port : 80 -> 80, 443 -> 443
      3. Development Tool
         1. Vite
         2. ESLint
         3. Visual Studio Code
   4. **3D Model Generation worker**
      1. Language : Python 3.13
      2. Development Tool
         1. PyCharm 2024.3.1.1
         2. Meshroom 2023.3.0
   5. **Image Validation server**
      1. Language : Python 3.13
      2. Development Tool
         1. PyCharm 2024.3.1.1
   6. **CI/CD**
      1. Docker 28.1.1
      2. Jenkins 2.504.1
         1. Port : 8000 -> 8080, 50000 -> 50000
   7. **Infra**
      1. MySQL
         1. version : 9.3.0
         2. port : 3306 -> 3306
      2. Redis
         1. version : 7.4.3
         2. port : 6379 -> 6379
      3. MinIO
         1. version : RELEASE.2025-03-12T18-04-18Z
         2. port : 8100 -> 9000, 8101 -> 9001
      4. Elasticsearch
         1. version : 8.7.1
         2. port : 8200 -> 9200
      5. Kibana
         1. version : 8.7.1
         2. port : 8601 -> 5601
      6. MongoDB
         1. version : 8.0.9
         2. port : 8300 -> 27017
      7. Prometheus
         1. version : 3.3.1
         2. port : 8500 -> 9090
      8. Grafana
         1. version : 12.0.0
         2. port : 8600 -> 3000
2. **빌드 시 사용되는 환경 변수 등의 내용 상세 기재**
   1. **Backend** 
      1. env.properties
         1. 경로
            1. 프로젝트 루트 디렉토리 (for test)
            2. Jenkins 마운트 볼륨 (for build)
         2. 항목
            1. DB\_DRIVER\_CLASS\_NAME
            2. DB\_URL
            3. DB\_USERNAME
            4. DB\_PASSWORD
            5. OAUTH\_REDIRECTURI\_PREFIX
            6. GOOGLE\_CLIENT\_ID
            7. GOOGLE\_CLIENT\_PASSWORD
            8. JWT\_SECRET
            9. LOGIN\_ENDPOINT
            10. REDIS\_HOST
            11. REDIS\_PWD
            12. MINIO\_URL
            13. MINIO\_ACCESS\_KEY
            14. MINIO\_SECRET\_KEY
            15. ELASTIC\_URI
            16. ELASTIC\_USERNAME
            17. ELASTIC PASSWORD
            18. ES\_TRUSTSTORE\_PATH
            19. ES\_TRUSTSTORE\_PASSWORD
            20. FRONT\_URL
            21. KEYSTORE\_PASSWORD
            22. SSL\_ENABLED
            23. MONGODB\_URL
            24. OPENAI\_API\_KEY
            25. OPENAI\_API\_URL
            26. OPENAI\_API\_MODEL
   2. **Backend(Chatting)**
      1. .env.properties-chat
         1. 경로
            1. 프로젝트 루트 디렉토리
         2. 항목
            1. JWT\_SECRET
            2. DB\_URL
            3. DB\_PORT
            4. DB\_USERNAME
            5. DB\_PASSWORD
            6. REDIS\_HOST
            7. REDIS\_PWD
            8. SSL\_PASSWORD
            9. SSL\_ENABLE
   3. **Frontend** 
      1. .env
         1. 경로
            1. 프로젝트 루트 디렉토리 (for test)
            2. Jenkins 마운트 볼륨 (for build)
         2. 항목

VITE\_API\_URL

* 1. Infra
     1. **MinIO**
        1. minio.env
           1. 경로

Jenkins 마운트 볼륨

* + - * 1. 항목

MINIO\_ROOT\_USER

MINIO\_ROOT\_PASSWORD

MINIO\_SERVER\_URL

MINIO\_BORWSER\_REDIRECT\_URL

TZ

* + 1. **Redis**
       1. redis.conf
          1. 경로

Jenkins 마운트 볼륨

* + - * 1. 수정사항

requirepass 수정

bind 주소 수정

* + 1. **MySQL**
       1. mysql.env
          1. 경로

Jenkins 마운트 볼륨

* + - * 1. 항목

MYSQL\_USER

MYSQL\_ROOT\_PASSWORD

MYSQL\_PASSWORD

TZ

* + 1. Elasticsearch, Kibana
       1. elk.env
          1. 경로

Jenkins 마운트 볼륨

* + - * 1. 항목

ELASTIC\_PASSWORD

KIBANA\_PASSWORD

STACK\_VERSION

CLUSTER\_NAME

ES\_PORT

KIBANA\_PORT

ES\_MEM\_LIMIT

KB\_MEM\_LIMIT

* + 1. Image validation server
       1. image.env
          1. 경로

jenkins 마운트 볼륨

* + - * 1. 항목

REDIS\_URL

REDIS\_PWD

REDIS\_STREAM\_IMAGE

REDIS\_CONSUMER\_GROUP

REDIS\_CONSUMER\_NAME

MINIO\_ENDPOINT

MINIO\_ACCESS\_KEY

MINIO\_SECRET\_KEY

MINIO\_SECURE

MINIO\_BUCKER

MINIO\_PUBLIC\_BASE

MYSQL\_HOST

MYSQL\_PORT

MYSQL\_DB

MYSQL\_USER

MYSQL\_PWD

MYSQL\_POOL\_SIZE

PREVIEW\_MAX\_SIZE

PREVIEW\_QUALITY

1. **배포 시 특이사항 기재**
   1. EC2
      1. Jenkins 공유 파일 디렉토리 생성
         1. ~/jenkins\_env
         2. ~/Jenkins\_home
      2. Docker 설치
         1. 설치
            1. curl –fsSL <https://get.docker.com> –o get-docker.sh
            2. sudo sh get-docker.sh
         2. 권한 부여
            1. sudo usermod –aG docker [username]
            2. sudo systemctl restart docker
            3. 로그아웃 및 재로그인
         3. Docker Volume 사전 생성
            1. elasticsearch-data
            2. elk-cert
            3. minio-data
            4. mysql-data
            5. nginx-certs
            6. redis-data
         4. Docker network 사전 생성
            1. $ docker network create –driver=bridge beet-network
         5. Docker context 설정 ( Jenkins 내에서 수행 )
            1. SSH 키 등록

ssh-add ~/.ssh/[등록할 ssh key.pem]

ssh-add –l

eval $(ssh-agent)

~/.ssh/config 파일 수정



docker context 생성

sudo docker context create [context이름] –docker host==ssh://[유저명]@[ip/도메인]

* + 1. Compose 파일
       1. docker-compose-be.yml



* + - 1. docker-compose-fe.yml



* + - 1. docker-compose-elk.yml



* + - 1. docker-compose-chat.yml



* + - 1. docker-compose-image.yml
      2. docker-compose-mongo.yml
      3. docker-compose-plg.yml
    1. Let’s Encrypt HTTPS(TLS) 인증서 발급
       1. EC2 3개에 대해 모두 발급
       2. sudo apt-get install letsencrypt –y
       3. certbot certonly –standalone –d {도메인주소}
       4. /etc/letsencrypt/live/{도메인주소}/ 경로에 다음 파일 확인
          1. fullchain.pem
          2. privkey.pem
    2. Jenkins
       1. Credentials 설정
          1. GitLab account
       2. Pipeline 생성
          1. beet-backend, beet-chat, beet-frontend, beet-image
          2. GitLab Webhook 설정, pipeline과 연동 (Jenkins Token 생성 필요)
          3. Pipeline script 작성

beet-backend



beet-frontend



beet-chat



beet-image



* 1. MinIO
     1. 컨테이너 실행 후 Web UI 접속
     2. Access Key 발급
     3. Spring Boot env.yml 파일에 토큰 추가
  2. MySQL
     1. timezone 설정
     2. 사용자 권한 부여
  3. 3D Modeling Server
     1. GPU 서버에 Meshroom 설치
     2. photogrammetry.mg 파이프라인 작성(기본 프리셋에 다음 옵션 추가)
        1. FeatureMatching Node : Minimal 2D Motion -> 2
        2. MeshFiltering : Smoothing Iterations -> 25
        3. Publish Node 추가, Texturing 노드의 산출물을 Publish Node에 연결하고 모델링 파일 저장 경로 지정
  4. HTTPS 설정
     1. 다음 인프라에 대해 HTTPS 보안 연결 설정
        1. backend server
        2. backend(chatting) server
        3. frontend server(nginx)
        4. MinIO
        5. Prometheus
        6. Grafana
        7. Kibana
        8. Elasticsearch <-> Kibana (자체 서명 사용)
        9. Elasticsearch <-> backend server (자체 서명 사용)

1. **DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록**
   1. .env
   2. .env.properties
   3. .env.properties-chat
   4. image.env
   5. keystorePassword.txt
   6. redis.conf
   7. elk.env
   8. mysql.env
   9. minio.env