

배포 설정

서울 공통 7반 새싹일기

(i10a706.p.ssafy.io)

2024.01.03 ~ 2024.02.16

이민주 김선영 김종인 김하은 윤선아 이민우

[Project stack skill version](#)

[EC2 포트 번호](#)

[환경 변수](#)

[Dockerfile](#)

[Jenkins 쉘 스크립트](#)

[NginX SSL 설정](#)

[NginX Reverse proxy 설정](#)

[MySQL 접속](#)

[redis 접속](#)

Project stack skill version

- Java : 17
- Spring boot : 3.1.2
- MySQL : 8.3.0
- Redis : 7.2.4
- Node.js : 21.5.0

EC2 포트 번호

- Back : 8081
- Front : 3000

- MySQL : 3306
- Redis : 6339
- Nginx : 80/443

환경 변수

- application.yml

backend/saessak/src/main/resources/application.yml

```
# MySQL8
spring:
  datasource:
    url: jdbc:mysql://i10a706.p.ssafy.io:3306/saessak?useSSL=false&characterEncoding=UTF-8&serverTimezone=UTC
    username: ssafytest
    password: saessak
    driver-class-name: com.mysql.cj.jdbc.Driver

# SQL
jpa:
  show-sql: true
  database-platform: org.hibernate.dialect.MySQL8Dialect
  hibernate:
    ddl-auto: update

data:
  redis:
    host: redis
    port: 6379

servlet:
  multipart:
    maxFileSize: 30MB
```

```

    maxRequestSize: 150MB

kakao:
  client:
    id: 8a0f1735f002f9571ca3a2c4e30db8da
    secret: wR3nvxNdDN0AMMBjsiznkFMGewPSwSql
  redirect:
    url: http://localhost:5173/

jwt:
  token:
    secret-key: dyAeHub00c8Ka0fYB6XEQoEj1QzRlVgtjNL8PYs1A1t
ymZvvqkcEU7L1imkKHeDa
  access-token:
    expire-length: 1209600000
  refresh-token:
    expire-length: 4838400000

cloud:
  aws:
    s3:
      bucket: saessack-photo-album
      credentials:
        access-key: AKIAUZRXT2PHNBJ4QE7T
        secret-key: 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
      region:
        static: ap-northeast-2
        auto: false
      stack:
        auto: false

OPENVIDU_URL: https://i10a706.p.ssafy.io:8443/
OPENVIDU_SECRET: admin

server:
  port: 8081

```

- saessak-63104-firebase-adminsdk-5s8na-3d340d35dd.json

backend/saessak/src/main/resources/saessak-63104-firebase-adminsdk-5s8na-3d340d35dd.json

```
{
  "type": "service_account",
  "project_id": "saessak-63104",
  "private_key_id": "3d340d35dd8a27ff21186ffe9e5181fdbadbbfdd",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQCSTEhuv/5XFjrp\nZz0zfi7p0GgFw73lmsHbSxtLS9l0PGHR7tWvcVaIvRgWB+T3Q6rn07VxkP8\n\nH/6q8bmfas0C4HgZtw0V2v9FdyQ6Wp4EgZEK6lazYayRSMPLt+IxMzynT\n\nFjZ0Lf\n4/YLuCaGfgRRMfaqyA7kGDE+/TuiLiQXjvkdGN+7chcJQtGHY\n\njrDskbqSbBVfc1\nnbZtlenf15DieYCWotqQEzqM60QsFWxH6EKMrANCJYc/\ny7vuDLjeq9G1bK5mm8Kij\nn9gfkVQBP6Zay9DS3t9T9NFVgiPwaanbp0FYu\n\nfSiBoywcSLMzwpt9t/G1gNYfNx/A\nnSZERGR7/AgMBAAECggEAAbvCyLyG3\n\nzJ6kkkLj8JkXRJmi+DbYmSJAoXReugm18xq\nnGzCw1Ydrw7IyL7eFVQjd/H\n\ntvf9ua7azK11ltEyR0lW0aJxGCM+QLLiQfgf/ybeHm\n\nnkC4LmxwDlleHyIh\n\n+a+jkTlFxLU/WPJVR0SX2nmSSgqz29ygw3xQc9Fx+FM1iSBo0\n\nn4t7a0YGM\n\nRw4vIwe56NV3zuVbxru0/kDBTF892y+nI10fWXHjbpcQ0NiHMLca+vK6\n\nnA\n\nu4+zdfYwGBGmIeP+UVemf0+zR7e9R9bqGnY42kRvHgT8JG1SGEsxjkwPMHd\n\n7qV0\nnWkgkXuN03TQiv7ARFrpnzLpha4YuaY+4xhF+CVds1QKBgQDXALg0d\n\n5jHYcTsy5Q3\nnDkAds7G05Q+u0dyQ69YlUN4dGygc9+2oh0bn3T/i5V0SGE\n\nKW3kJx1hK2Yq3jdix1\nnEL1bhcjbck3BT6wmmQfrvXtv+5a9BYBxr2JKRuz\n\ntxo4qpG/k33aKvNBh4pA8H1ET\nnEqdoSo+e5+arBgWRUy84Q1uCywKBgQDN\n\nJvF933UFqCPuTtLXfpD3RzbG7mfjeZFA\n\nnhdojbx+yuJVqngJay4DHQHYWm\n\n5j3i/jcS02LaNBexXR36r7WfK+G0P8GnS2oNcMj\n\nnOR7nEO8+mqez1GHE9r\n\nBQhoxSFda3pGjP3L4UbT9o070o07LPwpagzoz/MFpWrHP/\n\nnd0lGbhIqHQK\n\nBgQCwTfG/bHcGwGCiQ0Q4RWJE0083REKY+5Rf71Q7Y1XQDk3Ve2US\n\nnNHFY\n\nnvBdcfHg/a3VI0re885CaybmiCgqkuCEJW+wqnSB0YEUYZbCNC39hRR1ygY\n\nq\nns0PpXsG4tYIs64yp5KjDBIugSPY3pwKHnltSxRmCl+1qPtTf950YEnv/\n\nAwKBgQCY\nnSlJ2iE0Md/bETwtBIRloFxtYrV29hV8IMNTv+7cR6ugwDz/c0\n\nr1MMFP5EGSLTwLc\nnM97y4EneMGH3KvGjZ0h94e00+1sLRW2oVhZUls0jC5\n\nd2Pa4eUqor/NVpkMvVo1yT\nnAWCxNTAepkMGDrDsFZznnq0Y0sK1GKu7dse\n\nUWt1tGQKBgFZK2QizpPqQpIcSE7Gu\n\nn9bknzbf0qr4gjHDdN8TZUSgtTVNG\n\ntTo9YJXjl5p7IcdmwNYSp8RB8nNPQolEhQK1\n\nn+fv02164aSN72IvHxswP\n\n0/GSzg5ekpd5bALSA+QkMqPhDakw+NnzajiNSTBz2FGE\n\nn8WlthEpwmQV5A\n\nQoMmiicc/Rk\nn-----END PRIVATE KEY-----\n",
```

```

    "client_email": "firebase-adminsdk-5s8na@saessak-63104.iam.gserviceaccount.com",
    "client_id": "108220889093535431424",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-5s8na%40saessak-63104.iam.gserviceaccount.com",
    "universe_domain": "googleapis.com"
  }

```

- settings.ini

/recognition/settings.ini

```

[settings]
## s3 config
s3_access_key=AKIAUZRXT2PHA3NJ0076
s3_secret_key=phGc2EAFce5VaSlMBig8jiQ5sc0mUSLyPtD1l5bJ
s3_bucket_name=saessack-photo-album
s3_region_name=ap-northeast-2

## db config
db_dialect=mysql+mysqldb
db_user=user
db_password=password
db_url=localhost:3306
db_name=saessak

JWT_SECRET_KEY=dyAeHub00c8Ka0fYB6XEQoEj1QzRlVgtjNL8PYs1A1

```

- .env

/frontend/saessak

```

# API URL settings for PJT
VITE_VUE_API_URL=https://i10a706.p.ssafy.io/api

```

```

# aws setting
VITE_AWS_REGION=ap-northeast-2
VITE_AWS_ACCESS_KEY=AKIAUZRXT2PHA3NJ0076
VITE_AWS_SECRET_KEY=phGc2EAFce5VaSlMBig8jiQ5sc0mUSLyPtD1l5bJ
VITE_AWS_BUCKET_NAME=saessak-photo-album

# firebaseConfig
VITE_FIREBASE_APIKEY=AIzaSyCxxUbccZ7bwVgT7rzA7yjcLCXQPnvnmdC
VITE_FIREBASE_AUTHDOMAIN=saessak-63104.firebaseio.com
VITE_FIREBASE_PROJECTID=saessak-63104
VITE_FIREBASE_STORAGEBUCKET=saessak-63104.appspot.com
VITE_FIREBASE_MESSAGINGSENDERID=115340242931
VITE_FIREBASE_APPID=1:115340242931:web:390ff13b031281d17aac86
VITE_FIREBASE_MEASUREMENTID=G-PPTD28SK2P

# kakao
VITE_KAKAO_CLIENT_ID=8a0f1735f002f9571ca3a2c4e30db8da
VITE_KAKAO_CLIENT_SECRET=wR3nvxNdDN0AMMBjsiznkFMGewPSwSql
VITE_KAKAO_REDIRECT_URL=http://localhost:5173/

# openai
VITE_OPENAI_API_KEY=sk-nAuFdwf2Q5pm0VAXIRNuT3BlbkFJDxc4s6kjUA

```

- firebase-messaging-sw.js

```

importScripts('https://www.gstatic.com/firebasejs/8.10.0/firebase.js')
importScripts('https://www.gstatic.com/firebasejs/8.10.0/firebase-messaging.js')

const firebaseConfig = {
  apiKey: "AIzaSyCxxUbccZ7bwVgT7rzA7yjcLCXQPnvnmdC",
  authDomain: "saessak-63104.firebaseio.com",
  projectId: "saessak-63104",
  storageBucket: "saessak-63104.appspot.com",
  messagingSenderId: "115340242931",
  appId: "1:115340242931:web:390ff13b031281d17aac86",
  measurementId: "G-PPTD28SK2P"
};

```

```

firebase.initializeApp(firebaseConfig)
// Firebase Messaging 초기화
// Show Notification
// 메시지 기능 활성화를 알림
const messaging = firebase.messaging()

// 백그라운드
messaging.onBackgroundMessage((payload) => {
  console.log('background')
  console.log(payload)
  handleDataMessage(payload.data)
})

// 포그라운드
self.addEventListener('message', (event) => {
  if (event.data && event.data.type === 'foreground') {
    console.log('foreground')
    handleDataMessage(event.data.payload)
  }
})

function handleDataMessage(data) {
  console.log('데이터 메시지 처리')

  const notificationTitle = data.title
  const notificationOptions = {
    body: data.body
  }

  self.registration.showNotification(notificationTitle, notif.

}

// 클릭 이벤트 처리
self.addEventListener('notificationclick', (event) => {
  event.notification.close()
})

```

Dockerfile

- backend

```
FROM openjdk:17-alpine
ARG JAR_FILE=build/libs/*-SNAPSHOT.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

- frontend

```
FROM node:lts-alpine as build-stage

WORKDIR /app

COPY ./package*.json ./

RUN npm install
COPY . .

RUN npm run build

# production stage
FROM nginx:stable-alpine as production-stage
COPY --from=build-stage /app/dist /usr/share/nginx/html
RUN rm /etc/nginx/conf.d/default.conf

COPY ./nginx.conf /etc/nginx/conf.d

EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- ai


```

FROM python:3.8.18
COPY . /app

RUN apt-get update
RUN apt-get install ffmpeg libsm6 libxext6 -y
RUN apt-get install pkg-config libmariadb-dev-compat -y

RUN pip install -r /app/requirements.txt
RUN python3 /app/modules/download.py
WORKDIR /app

CMD ["gunicorn", "-w", "4", "-b", "0.0.0.0:5000", "deepface_s

```

Jenkins 쉘 스크립트

- backend

```

pipeline {
    agent any
    tools {
        gradle 'gradle'
        jdk 'jdk-17'
        dockerTool 'Docker'
    }
    stages {
        stage("Clear current directory"){
            steps{
                sh'''
                    rm -rf *
                '''
            }
        }
        stage('Git Clone') {

```

```

        steps {
            git branch: 'develop-be', url: 'https://lab.ssafty.com/s10-webmobile1-sub2/S10P12A706.git',
                credentialsId: 'gitlab-personal-token'
        }
    }
    stage('Apply application.yml files') {
        steps {
            withCredentials([file(credentialsId: 'application', variable: 'secretFile')]) {
                script {
                    sh 'mkdir -p backend/saessak/src/main/resources/'
                    sh 'cp $secretFile backend/saessak/src/main/resources/application.yml'
                }
            }
        }
    }
    stage('Apply firebase.json files') {
        steps {
            withCredentials([file(credentialsId: 'firebase-json', variable: 'secretFile')]) {
                script {
                    sh 'mkdir -p backend/saessak/src/main/resources/'
                    sh 'cp $secretFile backend/saessak/src/main/resources/saessak-63104-firebase-adminsdk-5s8na-3d340d35dd.json'
                }
            }
        }
    }
    stage('BE-Build') {
        steps {
            dir("./backend/saessak") {
                sh'''
                    gradle wrapper
                '''
            }
        }
    }
}

```

```

        chmod +x gradlew
        ./gradlew clean build -x test --sta
cktrace
        '''
    }
}
}
stage('Delete existing Docker images and container
s') {
    steps {
        sh'''
            if docker container inspect saessak_ser
ver >/dev/null 2>&1; then
                echo "container exists locally"
                docker stop saessak_server
                docker rm saessak_server
            else
                echo "container does not exist loca
lly"
            fi
            if docker image inspect server >/dev/nu
ll 2>&1; then
                echo "Image exists locally"
                docker rmi server
            else
                echo "Image does not exist locally"
            fi
        '''
    }
}
stage('Build and Deploy Docker') {
    steps {
        dir('./backend/saessak') {
            sh'''
                echo [BE] Build Docker Image!
                docker build -t server .
                echo [BE] Run Docker Container!
                docker run -dp 8081:8081 -e TZ=Asi

```

- ```

pipeline {
 agent any
 tools {
 nodejs 'nodejs'
 dockerTool 'Docker'
 }
 stages {
 stage('Clear current directory') {
 steps {
 sh'''
 rm -rf *
 '''
 }
 }
 stage('Git Clone') {
 steps {
 git url: 'https://lab.ssafy.com/s10-webmobile',
 branch: 'develop-fe',
 credentialsId: 'gitlab-personal-token'
 }
 }
 stage('FE-Build') {
 steps {
 dir('./frontend/saessak') {
 sh'''
 if docker container inspect saessak_c
 echo "container exists locally"
 '''
 }
 }
 }
 }
}

```

```

 docker stop saessak_client
 docker rm saessak_client
 else
 echo "container does not exist locally"
 fi
 if docker image inspect client >/dev/null
 echo "Image exists locally"
 docker rmi client
 else
 echo "Image does not exist locally"
 fi
 docker build -t saessak_client -f ./Dockerfile .
'''
}
}
}
stage('Build and Deploy Docker') {
 steps {
 dir('./frontend/saessak') {
 sh'''
 echo [FE] Run Docker Container!
 docker run -dp 5173:80 --name saessak_client saessak_client
 '''
 }
 }
}
}
}
}

```

- ai

```

pipeline {
 agent any
 tools {
 dockerTool 'Docker'
 }
 stages {
 stage("Clear current directory"){

```

```

 steps{
 sh'''
 rm -rf *
 '''
 }
 }
 stage('Git Clone') {
 steps {
 git branch: 'develop-ai', url: 'https://lab.ssafty.com/s10-webmobile1-sub2/S10P12A706.git',
 credentialsId: 'gitlab-personal-token'
 }
 }
 stage('Apply settings.ini files') {
 steps {
 withCredentials([file(credentialsId: 'settings', variable: 'secretFile')]) {
 script {
 sh 'mkdir -p recognition/'
 sh 'cp $secretFile recognition/settings.ini'
 }
 }
 }
 }
 stage('Delete existing Docker images and containers') {
 steps {
 sh'''
 if docker container inspect saessak_ai
 >/dev/null 2>&1; then
 echo "container exists locally"
 docker stop saessak_ai
 docker rm saessak_ai
 else
 echo "container does not exist locally"
 fi
 '''
 }
 }
}

```

```

 if docker image inspect ai >/dev/null 2
 >&1; then
 echo "Image exists locally"
 docker rmi ai
 else
 echo "Image does not exist locally"
 fi
 ...
}
}
stage('Build and Deploy Docker') {
 steps {
 dir('./recognition') {
 sh'''
 echo [AI] Build Docker Image!
 docker build -t ai .
 echo [AI] Run Docker Container!
 docker run -dp 5000:5000 -e TZ=Asia/Seoul --name saessak_ai --network saessak_server2 ai
 ...
 }
 }
}
}
}
}

```

## NginX SSL 설정

```

sudo snap install --classic certbot
sudo apt-add-repository -r ppa:certbot/certbot
sudo apt-get -y install python3-certbot-nginx

sudo certbot --nginx -d i10a706.p.ssafy.io

```

## NginX Reverse proxy 설정

/etc/nginx/sites-available/default.conf

```
server {
 listen 80;
 listen [::]:80;
 server_name i10a706.p.ssafy.io;

 client_max_body_size 100M;

 return 301 https://$host$request_uri;
}

server {
 listen 443 ssl default_server;
 listen [::]:443 ssl default_server;
 server_name i10a706.p.ssafy.io;

 client_max_body_size 100M;

 ssl_certificate /etc/letsencrypt/live/i10a706.p.ssafy.io/
 ssl_certificate_key /etc/letsencrypt/live/i10a706.p.ssafy
 include /etc/letsencrypt/options-ssl-nginx.conf;
 ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

 root /var/www/html;
 index index.html index.htm index.nginx-debian.html;

 location / {
 proxy_pass http://localhost:5173;
 }

 location /api {
 proxy_pass http://localhost:8081;
 }

 location /ai {
```



```

 proxy_pass http://localhost:5000;
 }

 location /chat {
 proxy_pass http://localhost:8081;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header Host $http_host;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_http_version 1.1;
 proxy_set_header Upgrade $http_upgrade;
 proxy_set_header Connection "Upgrade";
 proxy_set_header Accept-Encoding "";
 proxy_read_timeout 21600000;
 proxy_send_timeout 21600000;
 }
}

```

front container 내부 /etc/nginx/conf.d/nginx.conf

```

server {
 listen 80;
 listen [::]:80;
 server_name localhost;

 #access_log /var/log/nginx/host.access.log main;

 location / {
 root /usr/share/nginx/html;
 index index.html index.htm;
 try_files $uri /index.html;
 }

 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root /usr/share/nginx/html;
 }
}

```

---

## MySQL 접속

```
sudo docker exec -it mysql bash
```

```
$ mysql -u root -p
```

```
CREATE USER ssafytest@'%' identified by 'saessak';
GRANT ALL PRIVILEGES ON *.* TO ssafytest@'%';
FLUSH PRIVILEGES;
exit;
```

```
mysql -u ssafytest -p
```

```
create database if not exists saessak collate utf8mb4_general_ci;
```

```
docker run -d --name mysql-container -p 3306:3306 -v mysql-
volume:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=saessak mysql:
latest
```

---

## redis 접속

```
docker network create saessak_server2
```

```
docker run --name redis --network saessak_server2 -d redis
```

---

