프로젝트 배포 가이드

1. 개발 환경 정보

1.1 백엔드 (SpringBoot)

- JDK 버전: JDK 17
- SpringBoot 버전: 3.4.2
- Gradle 버전: 8.11.1
- IDE: IntelliJ IDEA 2023.3.8

1.2 백엔드 (FastAPI)

- Python 버전: 3.13.2
- FastAPI 버전: 0.115.8
- uvicorn 버전: 0.34.0
- IDE: VS Code

1.3 프론트엔드 (React)

- Node.js 버전
- npm 버전
- React 버전
- IDE: VS code

1.4 인프라 구성

- NginX
- Docker
- Jenkins

1.5 클라우드 환경

- AWS EC2
- AWS S3

- GitLab

2. 빌드 환경 설정

2.1 필수 환경 변수

```
# Spring Boot (.env) SPRING_DATASOURCE_URL=jdbc:mysql://172.17.0.5:3306/ski
SPRING_DATASOURCE_PASSWORD=dlswodmltnv1!
JWT_SECRET_KEY=6989816b247f2d01275210e92c61b4098af140a802e2886a487280e39eb3
AWS_ACCESS_KEY=AKIA4WJPW2CB5GGBXOEJ AWS_SECRET_KEY=EEdFs2XEiMLHmCCRt/zuPTYV
SPRING_DATASOURCE_URL=jdbc:mysql://172.17.0.5:3306/skinfit OPENAI_API_KEY=s
T0I3Bcu8jtUMld7rr4WHNEH2xEoJnMb-qJFT3BlbkFJ0gQKbmBvgeFSwmHZVve2mJR9VDg7_p8z
OCR_SECRET_KEY=QlNydkRieUZ2R0RsSFBWSE1VckVvVHd4aVNObGh3aXA= # React (.env)
name
```

2.2 빌드 프로파일

- dev: 로컬 개발환경
- mail: SMTP 설정
- oauth: 카카오 소셜로그인 설정
- prod: 운영 서버

3. Jenkins CI/CD 파이프라인

3.1 스프링부트

```
pipeline { agent any environment { DOCKERHUB_CREDENTIALS =
credentials('dockerhub-credentials') DOCKER_IMAGE = "kimdevspace/spring-
boot" VERSION = "latest" } stages { stage('Prepare Config Files') { steps
{ withCredentials([ file(credentialsId: 'application-prod-config',
variable: 'PROD_CONFIG'), file(credentialsId: 'application-mail-config',
variable: 'MAIL_CONFIG'), file(credentialsId: 'application-oauth-config',
variable: 'OAUTH_CONFIG') ]) { sh 'chmod -R 777
backend/src/main/resources' dir('backend/src/main/resources') { sh '''
echo "Copying Secret Config Files..." cp "$PROD_CONFIG" application-
prod.yml cp "$MAIL_CONFIG" application-mail.yml cp "$OAUTH_CONFIG"
application-oauth.yml ls -l ''' } } } } stage('Gradle Build') { steps {
dir('backend') { sh 'chmod +x gradlew' sh './gradlew clean build -
Dspring.profiles.active=prod' } } } stage('Docker Build') { steps {
dir('backend') { sh "docker build -t $DOCKER_IMAGE:$VERSION ." } } }
stage('Docker Login & Push') { steps { sh 'echo $DOCKERHUB_CREDENTIALS_PSW
| docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin' sh "docker
push $DOCKER_IMAGE:$VERSION" } } stage('Deploy') { steps { dir('backend')
{ sh 'chmod +x deploy.sh' sh './deploy.sh' } } } } }
```

리액트

```
// Jenkinsfile pipeline { agent any environment { DOCKERHUB_CREDENTIALS =
credentials('dockerhub-credentials') DOCKER_IMAGE = "kimdevspace/react-
app" VERSION = "latest" } stages { stage('React Build') { steps {
dir('frontend') { sh 'npm install' sh 'npm run build' } } } stage('Docker
Build') { steps { dir('frontend') { sh "docker build --build-arg
CACHEBUST=\$(date +%s) -t $DOCKER_IMAGE:$VERSION ." } } } stage('Docker
Login & Push') { steps { sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker
login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin' sh "docker push
$DOCKER_IMAGE:$VERSION" } } stage('Deploy') { steps { dir('frontend') { sh
'chmod +x deploy.sh' sh './deploy.sh' } } } } }
```

4. 주요 설정 파일

4.1 Nginx 설정

```
server { listen 80; listen [::]:80; server_name i12b111.p.ssafy.io; return 301 https://$host$request_uri; } server { listen 443 ssl; listen [::]:443 ssl; server_name i12b111.p.ssafy.io; ssl_certificate /etc/letsencrypt/live/i12b111.p.ssafy.io/fullchain.pem; ssl_certificate_key /etc/letsencrypt/live/i12b111.p.ssafy.io/privkey.pem; location /api { proxy_pass http://172.17.0.3:8080; proxy_set_header Host $host; proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme; } # FastAPI 엔드포인트 location /ocr { proxy_pass http://172.17.0.7:8000; # FastAPI 컨테이너의 IP와 포트 proxy_set_header Host $host; proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme; } location /recommend { proxy_pass http://172.17.0.7:8000; # FastAPI 컨테이너의 IP와 포트 proxy_set_header Host $host; proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme; } location / { root /var/www/html; index index.html; try_files $uri $uri/ /index.html; add_header Cache-Control "no-cache, no-store, must-revalidate"; add_header Pragma "no-cache"; add_header Expires 0; } }
```

4.2 프로퍼티 파일 목록

1. Spring Boot 설정
    - backend/src/main/resources/application.yml
    - backend/src/main/resources/application-dev.yml
    - backend/src/main/resources/application-mail.yml
    - backend/src/main/resources/application-oauth.yml
    - backend/src/main/resources/application-prod.yml

4.3 주요 계정 정보

1. AWS EC2 접속 키페어 : I12B111T.pem
2. Redis 비밀번호 : redis
3. Mysql 비밀번호 : dlswodmltnvb111
4. 젠킨스 계정 : ssafy12b1211, ssafy12b1211