



" 이 림은 내가 시작한거야 "

림특허청

목 차

I. 빌드 및 배포

- 1. 개발 환경 3
- 2. 설정 파일 및 환경 변수 정보..... 4
- 3. Frontend CI/CD17
- 4. Backend CI/CD18

II. 외부 서비스

- 1. 소셜 로그인 20

I. 빌드 및 배포

1. 개발환경

- **Server** : Ubuntu 20.04 LTS
- **JDK** : OpenJDK 11
- **Node.js** : 14.15.4
- **Docker** : 20.10.12
- **Nginx** : 1.23.3
- **MySQL** : 8.0.32
- **Jenkins** : 2.375.2
- **Openvidu** : 2.25.0

2. 설정 파일 및 환경 변수 정보

React

- application.yml (인증 서버)

```
server:
  port: 8070
  servlet:
    encoding:
      charset: UTF-8
      enabled: true
      force: true

spring:
  security:
    oauth2:
      client:
        registration:
          kakao:
            client-id: 클라이언트 아이디
            client-secret: 클라이언트 시크릿키
            client-authentication-method: post
            authorization-grant-type: authorization_code
            redirect-uri:
              https://j8a305.p.ssafy.io/api/auth/oauth2/callback/kakao
            scope:
              - profile_nickname
              - account_email
            client-name: Kakao

        provider:
          kakao:
            authorization-uri: https://kauth.kakao.com/oauth/authorize
            token-uri: https://kauth.kakao.com/oauth/token
            user-info-uri: https://kapi.kakao.com/v2/user/me
            user-name-attribute: id

app:
  server:
    user: https://j8a305.p.ssafy.io/api/mpoffice

  auth:
    token-expiration-msec: 864000000
    token-secret: 토큰 시크릿키

  oauth2:
    authorized-redirect-uris: https://j8a305.p.ssafy.io/oauth2/redirect,
    http://localhost:3000/oauth2/redirect,
    http://j8a305.p.ssafy.io:3000/oauth2/redirect
```

- application.yml (mpoffice 서비스 서버)

```
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url:
    username:
    password:

  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    generate-ddl: true
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        show_sql: false
        format_sql: true

    open-in-view: true
  servlet:
    multipart:
      max-file-size: 10MB
      max-request-size: 10MB
  redis:
    port:

    host:
      https://j8a305.p.ssafy.io

server:
  port:
  servlet:
    encoding:
      charset: UTF-8
      enabled: true
      force: true
```

```
cloud:  
  gcp:  
    storage:  
      bucket:  
      project-id:  
      credentials:  
        location: classpath:robust-metrics-380604-8e932ce5312a.json
```

```
app:  
  business:  
    url: https://j8a305.p.ssafy.io/api/auction  
  cors:  
    allowedOrigins: http://localhost:3000  
  auth:  
    token-expiration-msec:  
    token-secret:
```

- application.yml (auction 서비스 서버)

```
server:
  port:
  servlet:
    encoding:
      charset: UTF-8
      enabled: true
      force: true
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url:
    username:
    password:
  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    generate-ddl: true
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        show_sql: false
        format_sql: true
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
  jackson:
    deserialization:
      adjust-dates-to-context-time-zone: false
      time-zone: Asia/Seoul
  app:
    business:
      url: https://j8a305.p.ssafy.io/api/mpoffice
    cors:
      allowedOrigins: http://localhost:3000
```

- application.yml (auction 서비스 서버)

```
server:
  port: 8072
  servlet:
    encoding:
      charset: UTF-8
      enabled: true
      force: true
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://j8a305.p.ssafy.io:3072/auction-project
    username: a305
    password: dkanakf156346!!
  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    generate-ddl: true
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        show_sql: false
        format_sql: true
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
  jackson:
    deserialization:
      adjust-dates-to-context-time-zone: false
      time-zone: Asia/Seoul
  app:
    business:
      url: https://j8a305.p.ssafy.io/api/mpoffice
    cors:
      allowedOrigins: http://localhost:3000
```


Nginx

```
- App.conf
events{
    worker_connections 512;
}

http{
    upstream auth{
        server j8a305.p.ssafy.io:8070;
    }
    upstream mpoffice{
        server j8a305.p.ssafy.io:8071;
    }
    upstream auction{
        server j8a305.p.ssafy.io:8072;
    }
    upstream front{
        server j8a305.p.ssafy.io:3000;
    }
    server{
        listen 80;
        server_name j8a305.p.ssafy.io;

        return 308 https://j8a305.p.ssafy.io$request_uri;
    }

    server{
        listen 443 ssl;
        server_name j8a305.p.ssafy.io;

        ssl_certificate /etc/letsencrypt/live/j8a305.p.ssafy.io/fullchain.pem;
        ssl_certificate_key
        /etc/letsencrypt/live/j8a305.p.ssafy.io/privkey.pem;

        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 https;
        proxy_set_header X-Forwarded_Proto $scheme;
        proxy_headers_hash_bucket_size 512;
        proxy_redirect off;
```

Nginx

```
location /api/auth/ {
    proxy_pass http://auth;
}
location /api/mpoffice/ {
    proxy_pass http://mpoffice;
}
location /ws{
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_pass http://auction;
}

location /api/auction/ {
    proxy_pass http://auction;
}

location / {
    proxy_pass http://front;
}
}
```

Docker-compose.yml

version: "3.7"

services:

proxy-nginx:

user: root

container_name: nginx

image: nginx:latest

build:

context: .

dockerfile: ./Dockerfile

ports:

- 80:80

- 443:443

networks:

- a305-net

restart: always

volumes:

- /var/log/nginx:/var/log/nginx

- /etc/letsencrypt:/etc/letsencrypt

environment:

- TZ=Asia/Seoul

mysql-mpoffice:

container_name: mysql-mpoffice

image: mysql:8.0.31

volumes:

- a305-mysql-mpoffice-data:/var/lib/mysql-mpoffice

restart: always

ports:

- 3071:3306

Docker-compose.yml

```
environment:
  MYSQL_ROOT_PASSWORD: dkanakf156346!!
  MYSQL_DB: mpoffice-project
  MYSQL_USER: a305
  MYSQL_PASSWORD: dkanakf156346!!
networks:
  - a305-net

mysql-auction:
  container_name: mysql-auction
  image: mysql:8.0.31
  volumes:
    - a305-mysql-auction-data:/var/lib/mysql-auction-
  restart: always
  ports:
    - 3072:3306
  environment:
    TZ: Asia/Seoul
    MYSQL_ROOT_PASSWORD: dkanakf156346!!
    MYSQL_DB: auction-project
    MYSQL_USER: a305
    MYSQL_PASSWORD: dkanakf156346!!
  networks:
    - a305-net

volumes:
  a305-mysql-mpoffice-data:
  a305-mysql-auction-data:

networks:
  a305-net:
    external:
      name: a305-net
```

3. Frontend CI/CD

Jenkins

```

pipeline {
    agent any

    tools {
        nodejs "nodejs"
    }
    environment {
        DOCKER_REPOSITORY = "wx776654/a305"
        DOCKERHUB_CREDENTIALS = credentials('credential-username-and-
password-dockerhub')
    }

    stages {
        stage('Clone') {
            steps {
                git credentialsId: 'credential-username-and-password', branch:
                "develop/frontend", url: 'https://lab.ssafy.com/s08-blockchain-contract-
sub2/S08P22A305.git'
            }
        }
    }
//}

    stage('Docker Build') {
        steps{
            dir('Frontend') {
                sh 'docker build -t react-front:latest .'
            }
        }
    }

    stage('Deploy') {
        steps {
            sh 'docker rm -f front'
            sh 'docker run -d --name front -p 3000:3000 -u root react-front:latest'
        }
    }

    stage('Finish') {
        steps {
            sh 'docker images -qf dangling=true | xargs -l {} docker rmi {}'
        }
    }
}
    
```

4. Backend CI/CD

Jenkins

```

pipeline {
  agent any
  environment {
    DOCKER_REPOSITORY = "wx776654/a305"
    DOCKERHUB_CREDENTIALS = credentials('credential-username-
and-password-dockerhub')
  }

  stages {
    stage('Clone') {
      steps {
        git credentialsId: 'credential-username-and-password', branch:
"develop/mpoffice", url: 'https://lab.ssafy.com/s08-blockchain-contract-
sub2/S08P22A305.git'
      }
    }

    stage('Docker build'){
      steps{
        echo 'docker build'
        sh 'docker build -f mpoffice/Dockerfile -t
$DOCKER_REPOSITORY:springboot-mpoffice.'
      }
    }

    stage('Login') {
      steps {
        sh ""
        echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u

```

4. Backend CI/CD

Jenkins

```
$DOCKERHUB_CREDENTIALS_USR --password-stdin
    ""

}
}

stage('Push') {
    steps {
        sh "" docker push $DOCKER_REPOSITORY:springboot-mpoffice
    ""
    }
}

stage('Deploy') {
    steps {
        echo 'docker deploy'
        sh 'docker-compose up -d'
    }
}
}
}


post {
    always {
        sh 'docker logout'
    }
}
}
```

카카오 (Kakao)

1. <https://developers.kakao.com/> 로 접속한다
2. 내 애플리케이션 > 애플리케이션 추가하기 > 앱 이름 입력 > 사업자명 입력 > 저장
3. 좌측 Nav 바에서 앱 설정 > 요약 정보 > 앱 키 > REST API키 제품 설정 > 카카오 로그인 > 보안 > Client Secret의 코드 발급 기록해둡니다.

기본 정보

[수정](#)

앱 아이콘	
앱 이름	밈특허청
사업자명	밈특허청 주식회사

사용자가 카카오 로그인을 할 때 표시되는 정보입니다. 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

개인정보

항목 이름	ID	상태	
닉네임	profile_nickname	● 필수 동의	설정
프로필 사진	profile_image	● 필수 동의	설정
카카오계정(이메일)	account_email	● 필수 동의 [수집]	설정