포팅 메뉴얼

Environment

1. BackEnd
    - openjdk 11.0.20.1
    - spring boot 2.7.16
    - mysql 8.0.34
2. FrontEnd
    - node 18.17.0
    - react 18.2.0

Infra

Jenkins pipeline - BE

```
pipeline {
    agent {
        any {
            image 'gradle:jdk11'
            args '--network=host'

        }
    }

    environment {
        // 빌드 결과물이 저장될 경로
        BUILD_DIR = "/home/ubuntu/build"
        DOCKER_CONTAINER_NAME = "ccbb_back.app"
    }

    stages {
        stage('Git Clone') {
            steps {
                git branch: 'develop',
                credentialsId: 'ccbb_personal',
                url: 'https://lab.ssafy.com/s09-final/S09P31D104'
            }
        }

        stage('Build') {
            steps {
                // 스프링 프로젝트 빌드
                sh 'chmod +x /var/jenkins_home/workspace/ccbb_back/BE/ccbb/gradlew'
                sh 'cd /var/jenkins_home/workspace/ccbb_back/BE/ccbb/ && ./gradlew clean build -x test'
            }
        }

        stage('Deploy') {
            steps {
                sh "docker stop $DOCKER_CONTAINER_NAME || true"  // 기존 컨테이너가 없을 경우를 위해 무시
                sh "docker rm $DOCKER_CONTAINER_NAME || true"

                sh "mkdir -p $BUILD_DIR"

                sh "cp /var/jenkins_home/workspace/ccbb_back/BE/ccbb/build/libs/ccbb-0.0.1-SNAPSHOT.jar $BUILD_DIR"

                sh "cd $BUILD_DIR"
                sh "ls -al $BUILD_DIR"

                // 새로운 서버 실행
                //sh "docker run -d -e TZ=Asia/Seoul --name $DOCKER_CONTAINER_NAME --network=host -v $BUILD_DIR:/app -p 8081:8081 gradl
                sh "docker run -d -e TZ=Asia/Seoul --name $DOCKER_CONTAINER_NAME --network=host -v $BUILD_DIR:/app -v /home/ubuntu/spri

            }
        }
    }

    post {
        success {
            script {
                def Author_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()
                def Author_Name = sh(script: "git show -s --pretty=%ae", returnStdout: true).trim()
                mattermostSend (color: 'good',
                message: ":white_check_mark: BE 배포 성공: ${env.JOB_NAME} #${env.BUILD_NUMBER} by ${Author_ID}(${Author_Name})\n(<${env
                endpoint: 'https://meeting.ssafy.com/hooks/ky56pcia3fy37xknu49kj1kkro',
                channel: 'd104cicd'
                )
            }
        }
        failure {
            script {
                def Author_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()
                def Author_Name = sh(script: "git show -s --pretty=%ae", returnStdout: true).trim()
                mattermostSend (color: 'danger',
                message: ":x: BE 배포 실패: ${env.JOB_NAME} #${env.BUILD_NUMBER} by ${Author_ID}(${Author_Name})\n(<${env.BUILD_URL}|Deta
                endpoint: 'https://meeting.ssafy.com/hooks/ky56pcia3fy37xknu49kj1kkro',
                channel: 'd104cicd'
                )
            }
        }
    }

}
```

Jenkins pipeline - FE

```
pipeline {
    agent {
        any {
            images 'node:18'
            args '--network=host'

        }

    }

    environment {
        // 빌드 결과물이 저장될 경로
        BUILD_DIR = "/home/ubuntu/build"
        DOCKER_CONTAINER_NAME = "ccbb_front"
        CI = 'false'
    }

    stages {
        stage('Git Clone') {
            steps {
                git branch: 'develop',
                credentialsId: 'ccbb_personal',
                url: 'https://lab.ssafy.com/s09-final/S09P31D104'
            }
        }

        stage('Build') {
            steps {

                sh 'pwd'
                sh 'ls -al'

                nodejs(nodeJSInstallationName: 'NodeJS 18.17.0') {
                        sh 'cd /var/jenkins_home/workspace/ccbb_front/FE/ccbb && npm install && npm run build'
                }

                sh "mkdir -p $BUILD_DIR"
                sh "cp -r /var/jenkins_home/workspace/ccbb_front/FE/ccbb/build/* $BUILD_DIR"

            }
        }

        stage('Deploy') {
            steps {
                // 기존 서버 중지
                sh "docker stop $DOCKER_CONTAINER_NAME || true"
                sh "docker rm $DOCKER_CONTAINER_NAME || true"

                // 새로운 서버 실행
                sh "docker run -d -e TZ=Asia/Seoul --name $DOCKER_CONTAINER_NAME -p 3000:3000 --network=host -v $BUILD_DIR:/app node:1
            }
        }
    }

    post {
        success {
            script {
                def Author_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()
                def Author_Name = sh(script: "git show -s --pretty=%ae", returnStdout: true).trim()
                mattermostSend (color: 'good',
                message: ":white_check_mark: FE 배포 성공: ${env.JOB_NAME} #${env.BUILD_NUMBER} by ${Author_ID}(${Author_Name})\n(<${env
                endpoint: 'https://meeting.ssafy.com/hooks/ky56pcia3fy37xknu49kj1kkro',
                channel: 'd104cicd'
                )
            }
        }
        failure {
            script {
                def Author_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()
                def Author_Name = sh(script: "git show -s --pretty=%ae", returnStdout: true).trim()
                mattermostSend (color: 'danger',
                message: ":x: FE 배포 실패: ${env.JOB_NAME} #${env.BUILD_NUMBER} by ${Author_ID}(${Author_Name})\n(<${env.BUILD_URL}|Deta
                endpoint: 'https://meeting.ssafy.com/hooks/ky56pcia3fy37xknu49kj1kkro',
                channel: 'd104cicd'
                )
            }
        }
    }


}
```

Databse

```
CREATE DATABASE ccbb;
CREATE USER 'd104'@'%' IDENTIFIED BY 'ccbb';
GRANT ALL PRIVILEGES ON *.* FOR 'd104'@'*';
FLUSH PRIVILEGES;
```

외부 서비스

- Kakao API 키 발급
    - 카카오 로그인
    - 카카오 페이
- Riot API 키 발급
    - 티어 검색