

포팅 매뉴얼

배포 설정

▼ docker-compose.yml

```
services:
  be:
    container_name: be
    image: be
    build:
      context: ${BE_CONTEXT_PATH}
      dockerfile: ${METADATA}/be/Dockerfile
    ports:
      - "8080:8080"
    volumes:
      - ${BE_CONTEXT_PATH}:/app
    networks:
      - share-p-network
    environment:
      - TZ=Asia/Seoul
    depends_on:
      - mysql

  fe:
    container_name: fe
    image: fe
    build:
      context: ${FE_CONTEXT_PATH}
      dockerfile: ${METADATA}/fe/Dockerfile
    ports:
      - "5173:5173"
    volumes:
      - ${FE_CONTEXT_PATH}:/app
    environment:
      - TZ=Asia/Seoul
    networks:
      - share-p-network

  mysql:
    container_name: mysql
    image: mysql:latest
    ports:
      - "3305:3306"
    volumes:
      - mysql-volume:/var/lib/mysql
      - ~/init:/docker-entrypoint-initdb.d
    environment:
```

```

    MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
    TZ: Asia/Seoul
networks:
  - share-p-network
restart: unless-stopped

volumes:
  mysql-volume:

networks:
  share-p-network:

```

▼ .env

```

JENKINS_WORKSPACE=/var/lib/jenkins/workspace
FE_CONTEXT_PATH=${JENKINS_WORKSPACE}/share-p_main/fe
BE_CONTEXT_PATH=${JENKINS_WORKSPACE}/share-p_main/be

METADATA=${JENKINS_WORKSPACE}/share-p

MYSQL_ROOT_PASSWORD=4YRxQjNH7hNsdjeXBqQx6iN6l4aqpe0j

```

▼ Jenkinsfile

```

pipeline {
  agent any

  environment {
    METADATA = '/var/lib/jenkins/workspace/share-p'

    SRC_ENV = './fe'
    SRC_RESOURCES = './be/src/main/resources'
  }

  tools {
    gradle 'default'
    nodejs 'default'
  }

  stages {
    stage('파일 복사') {
      when {
        branch 'main'
      }
      steps {
        sh "rm -f ${env.SRC_RESOURCES}/application.yml && mkdir ${env.SRC_RES

```

```

OURCES} || true"
    sh "cp ${env.METADATA}/be/application.yml ${env.SRC_RESOURCES}/applic
ation.yml"

    sh "rm -f ${env.SRC_ENV}/.env"
    sh "cp ${env.METADATA}/fe/.env ${env.SRC_ENV}/.env"
  }
}

stage('빌드') {
  when {
    branch 'main'
  }
  parallel {
    stage("Spring Boot App 빌드") {
      steps {
        dir('be') {
          /** clean and slow build with info **/
          // sh "chmod +x gradlew && ./gradlew clean --info build"

          /** normal build **/
          // sh "chmod +x gradlew && ./gradlew build"

          /** fast build **/
          sh 'chmod +x gradlew && ./gradlew bootJar'
        }
      }
    }

    stage('React App 빌드') {
      steps {
        dir('fe') {
          sh 'npm i && npm run build'
        }
      }
    }
  }
}

stage('Container 재시작') {
  when {
    branch 'main'
  }
  steps {
    sh """
    services=$(docker compose -f ${env.METADATA}/docker-compose.yml co
nfig --services | grep -E '^be|^fe')
    docker compose -f ${env.METADATA}/docker-compose.yml restart \${serv
ices}
    """
    // sh 'docker compose -f ${env.METADATA}/docker-compose.yml restart f

```

```

e $(docker compose -f ${env.METADATA}/docker-compose.yml config --services |
grep \'^be\')'
    // sh 'docker compose -f ${env.METADATA}/docker-compose.yml restart f
e $(docker compose -f ${env.METADATA}/docker-compose.yml config --services |
grep \'^be\')'

    }
  }
}

post {
  success {
    script {
      def authorName = sh(script: 'git show -s --pretty=%an', returnStdout:
true).trim()
      def commiterName = sh(script: 'git show -s --pretty=%cn', returnStdou
t: true).trim()

      mattermostSend(
        color: 'good',
        message: "${env.BRANCH_NAME}: ${commiterName}님이 커밋하고 ${authorNam
e}님이 요청한 ${env.BUILD_NUMBER}번째 Merge 나왔습니다~ (<${env.BUILD_URL}|상세정보
>)",
        endpoint: 'https://meeting.ssafy.com/hooks/4j87ayp3g7db9pg3nzun1efr
dw',
        icon: 'https://www.notion.so/image/https%3A%2F%2Fcdn.icon-icons.co
m%2Ficons%2F2699%2FPNG%2F512%2Fjenkins_logo_icon_170552.png?table=block&id=5
75d933c-a155-4d41-987e-887b341928ba&spaceId=0f7cf07a-b632-46e9-b39b-7c12c129b
0d0&userId=fcb99ecb-2346-4edb-9302-eea7d618e9d4&cache=v2',
        channel: 'b207-jenkins'
      )
    }
  }

  failure {
    script {
      def authorName = sh(script: 'git show -s --pretty=%an', returnStdout:
true).trim()
      def commiterName = sh(script: 'git show -s --pretty=%cn', returnStdou
t: true).trim()

      mattermostSend(
        color: 'danger',
        message: "${env.BRANCH_NAME}: ${commiterName}님이 커밋하고 ${authorNam
e}님이 요청한 ${env.BUILD_NUMBER}번째 Merge 아직 안 나왔습니다. (<${env.BUILD_URL}|
상세정보>)",
        endpoint: 'https://meeting.ssafy.com/hooks/4j87ayp3g7db9pg3nzun1efr
dw',
        icon: 'https://www.notion.so/image/https%3A%2F%2Fcdn.icon-icons.co
m%2Ficons%2F2699%2FPNG%2F512%2Fjenkins_logo_icon_170552.png?table=block&id=5
75d933c-a155-4d41-987e-887b341928ba&spaceId=0f7cf07a-b632-46e9-b39b-7c12c129b

```

```

0d0&userId=fcb99ecb-2346-4edb-9302-eea7d618e9d4&cache=v2',
    channel: 'b207-jenkins'
  )
}
}
}
}
}

```

▼ be/Dockerfile

```

FROM openjdk:17-jdk-slim

WORKDIR /app

ENTRYPOINT ["java", "-jar", "build/libs/be-0.0.1-SNAPSHOT.jar"]

```

▼ be/init/init.sql

```

create database if not exists sharep;

```

▼ be/application.yml

```

server:
  port: 8080
  servlet:
    context-path: /api
    encoding:
      charset: UTF-8
      enabled: true
      force: true

spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://mysql:3306/sharep?&useSSL=false&allowPublicKeyRetrieval
    =true&useUnicode=true&serverTimezone=Asia/Seoul
    username: root
    password: 4YRxQjNH7hNsdjeXBqQx6iN6l4aqpe0j
    hikari:
      maximum-pool-size: 20
      leak-detection-threshold: 3000

  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MySQLDialect
        format_sql: false
        show_sql: true

```

```

    open-in-view: false

jwt:
  header: Authorization
  expiration_time: 86400000
  secret: dltmdalsWKdWKdasdfkj2349uwkbzkoilefjlkfwfwfwefwef11

cloud:
  aws:
    s3:
      bucket: share-p
      stack.auto: false
      region.static: ap-northeast-2
      credentials:
        accessKey: AKIAXYKJWQPALOR2ACD5
        secretKey: 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

cors:
  allowedOrigins: http://localhost:5173,https://share-p.com
  allowedMethods: GET,POST,PUT,DELETE,PATCH,OPTIONS,HEAD
  allowedHeaders: Content-Type,Authorization
  exposedHeaders: '*'
  allowCredentials: false
  maxAge: 3600

schedule:
  crone: 0 5 18 * * 1-6 # 초 분 시 일 월 요일(월~금)

openai:
  url: https://api.openai.com/v1/chat/completions
  model: gpt-3.5-turbo-0125
  secret_key: sk-proj-1zVWvl4XbJEX9yoRXtaRT3BlbkFJVQiFd9qmnoniABEEr1Yb
  max_token: 5
  prompt: '당신은 작업을 빠르게 검토하고 요점을 추출하는 요약 전문가입니다. 작업들 중 핵심 사항을 추출하여 산문 형식의 30글자 이내의 한문장의 JSON 형태로 제공해주세요.'

[예시문]
{
  "result" : "라우트, 스타일 개편 및 Gantt 차트, dayjs 추가"
}

[작업]'

notification:
  mattermost:
    enabled: false # mmSender를 사용할 지 여부, false면 알림이 오지 않는다
    webhook-url: https://meeting.ssafty.com/hooks/rofzj1w5o78jfqte9b7jd5539y#
    위의 Webhook URL을 기입
    channel: # 기본 설정한 채널이 아닌 다른 채널로 보내고 싶을 때 기입한다
    pretext: # attachments의 상단에 나오게 되는 일반 텍스트 문자
    color: # attachment에 왼쪽 사이드 컬러. default=red

```

```
author-name: # attachment의 상단에 나오는 이름
author-icon: # author-icon 왼쪽에 나올 아이콘의 url링크
footer:
```

```
logging:
  level:
    com.zaxxer.hikari.HikariConfig: DEBUG
    com.zaxxer.hikari: TRACE
```

▼ fe/Dockerfile (React)

```
FROM node:20.11.0

WORKDIR /app

ENTRYPOINT ["npm", "run", "start", "--", "http-server", "-p", "5173", "-P", "http://localhost:5173?"]
```

▼ fe/.env

```
VITE_END_POINT=https://share-p.com/api
```

▼ NginX config

```
upstream jenkins {
    keepalive 32; # keepalive connections
    server 127.0.0.1:9091; # jenkins ip and port
}

upstream be {
    server 127.0.0.1:8080;
    # server 127.0.0.1:8081;
    # server 127.0.0.1:8082;
    # server 127.0.0.1:8083;
    # server 127.0.0.1:8084;
}

upstream fe {
    server 127.0.0.1:5173;
}

# Required for Jenkins websocket agents
map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

map $host $redirect {
```

```

"share-p.com" https://share-p.com$request_uri;
"k10b207.p.ssafy.io" https://share-p.com$request_uri;
"3.38.180.168" https://share-p.com$request_uri;
}

### jenkins server ###
server {

    server_name      share-p.com;

    listen           9090 ssl;          # Listen on port 80 for IPv4 requests

        # 도메인을 변경 했을 때, 인증서에 도메인을 추가해주어야 합니다.
    # 아래는 share-p.com 도메인에 대해 발급한 인증서입니다.
    ssl_certificate   /etc/letsencrypt/live/share-p.com/fullchain.pem;
# managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/share-p.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    root             /var/run/jenkins/war/;

    access_log        /var/log/nginx/jenkins.access.log;
    error_log         /var/log/nginx/jenkins.error.log;

        # pass through headers from Jenkins that Nginx considers invalid
    ignore_invalid_headers off;

    location ~ "^/static/[0-9a-fA-F]{8}\\/(.*)$" {
        # rewrite all static files into requests to the root
        # E.g /static/12345678/css/something.css will become /css/something.css
        rewrite "^/static/[0-9a-fA-F]{8}\\/(.*)$" /$1 last;
    }

    location /userContent {
        # have nginx handle all the static requests to userContent folder
        # note : This is the $JENKINS_HOME dir
        root /var/lib/jenkins/;
        if (!-f $request_filename){
            # this file does not exist, might be a directory or a /**view**
            rewrite (.*?) /$1 last;
            break;
        }
        sendfile on;
    }
}

```



```

        location / {
            sendfile off;
            proxy_pass      http://jenkins;
            proxy_redirect   off;
            proxy_http_version 1.1;

            # Required for Jenkins websocket agents
            proxy_set_header    Connection      $connection_upgrade;
            proxy_set_header    Upgrade         $http_upgrade;

            proxy_set_header    Host            $http_host;
            proxy_set_header    X-Real-IP       $remote_addr;
            proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;

r;

            proxy_set_header    X-Forwarded-Proto $scheme;
            proxy_max_temp_file_size 0;

            #this is the maximum upload size
            client_max_body_size      10m;
            client_body_buffer_size    128k;

            proxy_connect_timeout      90;
            proxy_send_timeout          90;
            proxy_read_timeout          90;
            proxy_request_buffering     off; # Required for HTTP CLI command
s
        }

        error_page 497 https://share-p.com:9090$request_uri;

    }

server {
    server_name      share-p.com;

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot

    ssl_certificate /etc/letsencrypt/live/share-p.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/share-p.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    location / {
        proxy_pass http://fe;

        proxy_set_header    Host            $http_host;
        proxy_set_header    X-Real-IP       $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;

```

```

or;
    proxy_set_header    X-Forwarded-Proto $scheme;
}

location /api {
    proxy_pass http://be;

    proxy_set_header    Host                $http_host;
    proxy_set_header    X-Real-IP           $remote_addr;
    proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_f
or;
    proxy_set_header    X-Forwarded-Proto $scheme;

    proxy_connect_timeout    300;
    proxy_send_timeout       300;
    proxy_read_timeout       300;
#    proxy_set_header Origin "";
    proxy_set_header Connection "";
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_buffering off;
    proxy_cache off;
    chunked_transfer_encoding off;

    #proxy_set_header Connection "upgrade";

}

error_page 497 https://share-p.com$request_uri;
}

server {
    server_name k10b207.p.ssafy.io;

    return 301 https://share-p.com$request_uri;
}

```

▼ ufw

```

$ sudo ufw status
Status: active

```

To	Action	From
--	-----	----
22	ALLOW	Anywhere
443	ALLOW	Anywhere
2222	ALLOW	Anywhere
9091/tcp	ALLOW	14.50.47.221
9091/tcp	ALLOW	52.0.0.0/8
80	ALLOW	Anywhere
9091/tcp	ALLOW	221.168.252.17

6100	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
2222 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
6100 (v6)	ALLOW	Anywhere (v6)