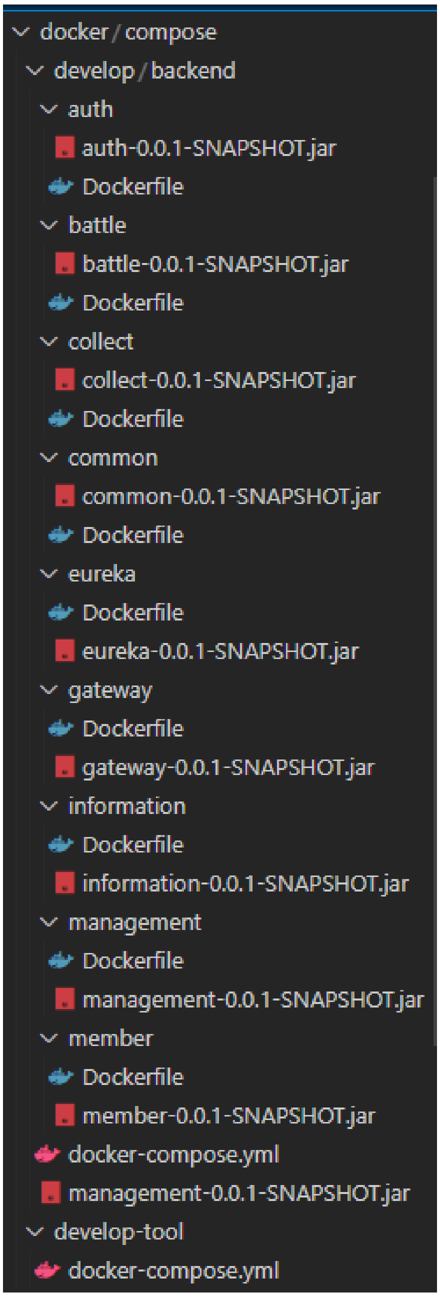
⛅

포팅 메뉴얼

tags	인프라
상태	Not started
프론트 가이드	

■도커파일

도커파일 구조



```
docker / compose
  develop / backend
    auth
      auth-0.0.1-SNAPSHOT.jar
      Dockerfile
    battle
      battle-0.0.1-SNAPSHOT.jar
      Dockerfile
    collect
      collect-0.0.1-SNAPSHOT.jar
      Dockerfile
    common
      common-0.0.1-SNAPSHOT.jar
      Dockerfile
    eureka
      Dockerfile
      eureka-0.0.1-SNAPSHOT.jar
    gateway
      Dockerfile
      gateway-0.0.1-SNAPSHOT.jar
    information
      Dockerfile
      information-0.0.1-SNAPSHOT.jar
    management
      Dockerfile
      management-0.0.1-SNAPSHOT.jar
    member
      Dockerfile
      member-0.0.1-SNAPSHOT.jar
    docker-compose.yml
    management-0.0.1-SNAPSHOT.jar
  develop-tool
    docker-compose.yml
```

back-end 도커파일

```
# auth Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=auth-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# battle Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=battle-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# collect Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=collect-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# common Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=common-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# eureka Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=eureka-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# gateway Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=gateway-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# information Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=information-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# management Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=management-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

```
# member Dockerfile
FROM openjdk:11-jdk
ARG JAR_FILE=member-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} ./app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

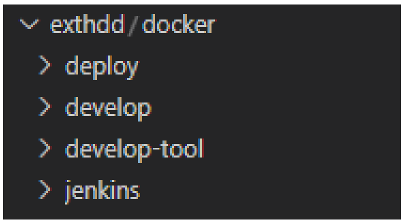
back-end docker-compose.yml

```
version: "3"

services:
  eureka:
    container_name: eureka
    environment:
      - TZ=Asia/Seoul
    # Dockerfile이 있는 위치
    build: ./eureka
    ports:
      - "8761:8761"
    networks:
      - develop-net
      - develop-tool-net

  gateway:
    container_name: gateway
    environment:
      - TZ=Asia/Seoul
    build: ./gateway
    ports:
      - "8080:8080"
    networks:
      - develop-net
      - develop-tool-net

  member:
    container_name: member
    environment:
      - TZ=Asia/Seoul
    build: ./member
    networks:
      - develop-net
      - develop-tool-net

  auth:
    container_name: auth
    environment:
      - TZ=Asia/Seoul
    build: ./auth
    networks:
      - develop-net
      - develop-tool-net

  common:
    container_name: common
    environment:
      - TZ=Asia/Seoul
    build: ./common
    networks:
      - develop-net
      - develop-tool-net

  information:
    container_name: information
    environment:
      - TZ=Asia/Seoul
    build: ./information
    networks:
      - develop-net
      - develop-tool-net

  management:
    container_name: management
    environment:
      - TZ=Asia/Seoul
    build: ./management
    networks:
      - develop-net
      - develop-tool-net

  collect:
    container_name: collect
    environment:
      - TZ=Asia/Seoul
    build: ./collect
    networks:
      - develop-net
      - develop-tool-net
  battle:
    container_name: battle
    environment:
      - TZ=Asia/Seoul
    build: ./battle
    ports:
      - "11050:11050"
    networks:
      - develop-net
      - develop-tool-net

networks:
  develop-net:
    name: develop-net
  develop-tool-net:
    external: true
    name: develop-tool-net
```

develop-tool docker-compose.yml

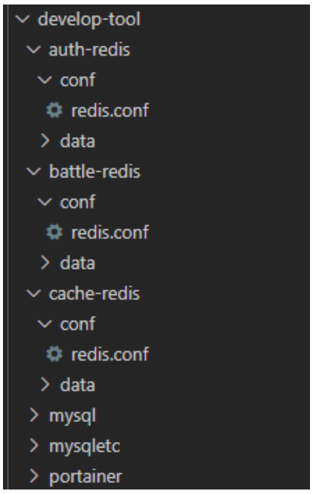
```
version: "3"

services:
  mysql:
    container_name: mysql
    image: mysql
    ports:
      - 3306:3306
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=paymong103.!
      - TZ=Asia/Seoul
    command:
      - --character-set-server=utf8mb4
      - --collation-server=utf8mb4_unicode_ci
      - --max_connections=300
    volumes:
      - /exthdd/docker/develop-tool/mysql:/var/lib/mysql
      - /exthdd/docker/develop-tool/mysqletc/conf.d:/etc/mysql/conf.d
    networks:
      - develop-tool-net

  auth-redis:
    container_name: auth-redis
    image: redis:7.0.8
    ports:
      - 6379:6379
    restart: always
    volumes:
      - /exthdd/docker/develop-tool/auth-redis/conf/redis.conf:/etc/redis/redis.conf
      - /exthdd/docker/develop-tool/auth-redis/data:/data
    command: redis-server /etc/redis/redis.conf
    networks:
      - develop-tool-net

  battle-redis:
    container_name: battle-redis
    image: redis:7.0.8
    ports:
      - 6380:6379
    restart: always
    volumes:
      - /exthdd/docker/develop-tool/battle-redis/conf/redis.conf:/etc/redis/redis.conf
      - /exthdd/docker/develop-tool/battle-redis/data:/data
    command: redis-server /etc/redis/redis.conf
    networks:
      - develop-tool-net

  cache-redis:
    container_name: cache-redis
    image: redis:7.0.8
    ports:
      - 6381:6379
    restart: always
    volumes:
      - /exthdd/docker/develop-tool/cache-redis/conf/redis.conf:/etc/redis/redis.conf
      - /exthdd/docker/develop-tool/cache-redis/data:/data
    command: redis-server /etc/redis/redis.conf
    networks:
      - develop-tool-net

networks:
  develop-tool-net:
    name: develop-tool-net
```

■volume



```
exthdd / docker
  deploy
  develop
  develop-tool
  jenkins
```

develop-tool volume

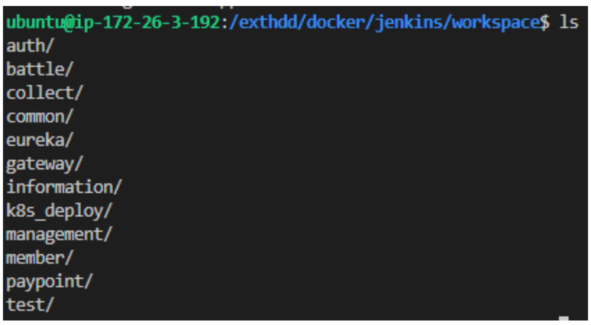


```
develop-tool
  auth-redis
    conf
      redis.conf
    data
  battle-redis
    conf
      redis.conf
    data
  cache-redis
    conf
      redis.conf
    data
  mysql
  mysqletc
  portainer
```

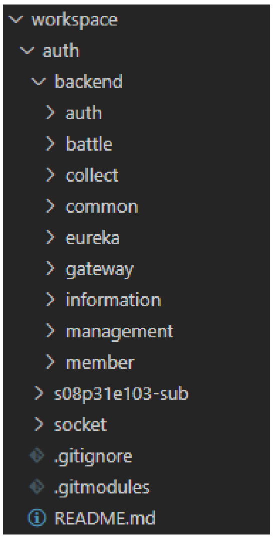
```
bind 0.0.0.0
port 6379
requirepass paymong103.!
```

모든 레디스 redis.conf 파일 설정

jenkins



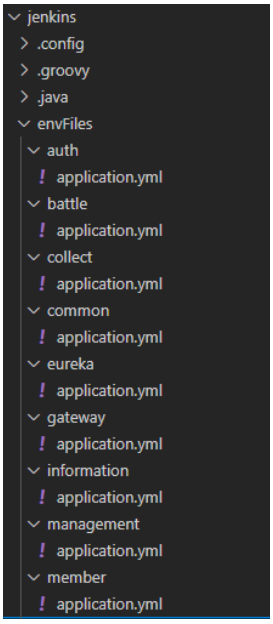
```
ubuntu@ip-172-26-3-192:/exthdd/docker/jenkins/workspace$ ls
auth/
battle/
collect/
common/
eureka/
gateway/
information/
k8s_deploy/
management/
member/
paypoint/
test/
```



```
workspace
  auth
    backend
      auth
      battle
      collect
      common
      eureka
      gateway
      information
      management
      member
    s08p31e103-sub
    socket
    .gitignore
    .gitmodules
    README.md
```

💡 jenkins로 깃랩 브랜치 웹훅으로 clone받은 후 build해 $service/build/libs 디렉토리에 jar 파일 생성
생성된 jar파일 home/ubuntu/docker/compose/develop/backend/$service로 Transefer 이후 도커 컨테이너 실행

env file



```
jenkins
  .config
  .groovy
  .java
  envFiles
    auth
      application.yml
    battle
      application.yml
    collect
      application.yml
    common
      application.yml
    eureka
      application.yml
    gateway
      application.yml
    information
      application.yml
    management
      application.yml
    member
      application.yml
```

auth application.yml

```
server:
  port: 0

spring:
  application:
    name: auth
  datasource:
    url: jdbc:mysql://mysql:3306/auth?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver

  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        show_sql: true
        format_sql: true
        use_sql_comments: true
  redis:
    host: k8e103.p.ssafy.io
    port: 6379
    password: paymong103.!


eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
        defaultZone: http://eureka:8761/eureka

logging:
  level:
    org:
      hibernate: info

header:
  member: "MemberId"
  mong: "MongId"


jwt:
  secret: eyJzdWIiOiJtZXRhbW9uZyIsIm5hbWUiOiJwYXltb25nIiwiaWF0IjoxMDN9eyJzdWIiOiJtZXRhbW9uZyIsIm5hbWUiOiJwYXltb25nIiwiaWF0IjoxMDN
```

battle application.yml

```
server:
  port: 11050

spring:
  application:
    name: battle
  redis:
    matching:
      host: k8e103.p.ssafy.io
      port: 6380
      password: paymong103.!

battle:
  max_distance: 300
  max_search: 10
  max_turn: 10
  default_health: 500

eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka
```

collection application.yml

```
server:
  port: 0

spring:
  application:
    name: collect
  datasource:
    url: jdbc:mysql://mysql:3306/collect?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver

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

eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka

header:
  member: "MemberId"
  mong: "MongId"

logging:
  level:
    org:
      hibernate: info
```

common application.yml

```
server:
  port: 0

spring:
  application:
    name: common
  datasource:
    url: jdbc:mysql://mysql:3306/common?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver

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

eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka

logging:
  level:
    org:
      hibernate: info
```

eureka application.yml

```
server:
  port: 8761

spring:
  application:
    name: discovery

eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
  server:
    eviction-interval-timer-in-ms: 10000

management:
  endpoint:
    shutdown:
      enabled: true
  endpoints:
    web:
      exposure:
        include: shutdown
```

gateway application.yml

```
server:
  port: 8080

spring:
  application:
    name: gateway
  datasource:
    url: jdbc:mysql://mysql:3306/management?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver
  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        show_sql: true
        format_sql: true
        use_sql_comments: true
  cloud:
    gateway:
      default-filters:
        - name: GlobalFilter
          args:
            baseMessage: Hello Spring Cloud Gateway Global Filter
            preLogger: true
            postLogger: true
      routes:
        - id: auth
          predicates:
            - Path=/auth/**
          uri: lb://AUTH
        - id: collect
```

```
          predicates:
            - Path=/collect/**
          uri: lb://COLLECT
          filters:
            - CustomFilter
        - id: common
          predicates:
            - Path=/common/**
          uri: lb://COMMON
          filters:
            - CustomFilter
        - id: information
          predicates:
            - Path=/information/**
          uri: lb://INFORMATION
          filters:
            - CustomFilter
        - id: management
          predicates:
            - Path=/management/**
          uri: lb://MANAGEMENT
          filters:
            - CustomFilter
        - id: member
          predicates:
            - Path=/member/**
          uri: lb://MEMBER
          filters:
            - CustomFilter
  redis:
    host: k8e103.p.ssafy.io
    port: 6379
    password: paymong103.!


eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka

management:
  endpoint:
    shutdown:
      enabled: true
  endpoints:
    web:
      exposure:
        include: shutdown

jwt:
  secret: eyJzdWIiOiJtZXRhbW9uZyIsIm5hbWUiOiJwYXltb25nIiwiaWF0IjoxMDN9eyJzdWIiOiJtZXRhbW9uZyIsIm5hbWUiOiJwYXltb25nIiwiaWF0IjoxMDN
```

information application.yml

```
server:
  port: 0

spring:
  application:
    name: information
  datasource:
    url: jdbc:mysql://mysql:3306/management?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver

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

eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka

logging:
  level:
    org:
      hibernate: info
header:
  member: "MemberId"
  mong: "MongId"
```

management application.yml

```
server:
  port: 0

spring:
  application:
    name: management
  datasource:
    url: jdbc:mysql://mysql:3306/management?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver
  redis:
    port: 6381
    host: dev.paymong.com
    password: paymong103.!

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

eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka

logging:
  level:
    org:
      hibernate: info

header:
  member: "MemberId"
  mong: "MongId"
```

member application.yml

```
server:
  port: 0

spring:
  application:
    name: member
  datasource:
    url: jdbc:mysql://mysql:3306/member?useSSL=false&useUnicode=true&serverTimezone=Asia/Seoul
    username: paymong103
    password: paymong103.!
    driver-class-name: com.mysql.cj.jdbc.Driver

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

eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://eureka:8761/eureka

header:
  member: "MemberId"
  mong: "MongId"

logging:
  level:
    org:
      hibernate: info
```