## **CICD**

```
Dockerfile
   discovery
   gateway
   auth
   member
   book
   donation
   chat
   notification
   develop-net docker-compose.yml
   develop-tool-net docker-compose.yml
   develop-kafka-net docker-compose.yml
   docker-compose-ui.yml
application.yml
   discovery
   gateway
   auth
   member
   book
   donation
   Chat
   notification
pipeline
```

## **Dockerfile**

directory

```
ubuntu@ip-172-26-0-253:~/docker$ ls auth develop-tool docker-compose.yml gateway book discovery donation member ubuntu@ip-172-26-0-253:~/docker$
```

## discovery

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

## gateway

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

## auth

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

### member

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

### book

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

### donation

```
FROM openjdk:11-jdk

ARG JAR_FILE=donation-0.0.1-SNAPSHOT.jar

COPY ${JAR_FILE} ./app.jar

ENTRYPOINT ["java","-jar","/app.jar"]
```

## chat

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

### notification

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

## develop-net docker-compose.yml

```
version: "3"
services:
 discovery:
   container_name: discovery
    environment:
     - TZ=Asia/Seoul
    build: ./discovery
    ports:
     - 8761:8761
    networks:
     - develop-net
     - develop-tool-net
     - kafka-network
  gateway:
   container_name: gateway
    environment:
     - TZ=Asia/Seoul
    build: ./gateway
    ports:
     - 8000:8000
    networks:
     - develop-net
     - develop-tool-net
     - kafka-network
  auth:
```

```
container_name: auth
    \hbox{\it environment:}
      - TZ=Asia/Seoul
    build: ./auth
    networks:
      - develop-net
      - develop-tool-net
      - kafka-network
  member:
    container_name: member
    environment:
      - TZ=Asia/Seoul
    build: ./member
    networks:
     - develop-net
      - develop-tool-net
      - kafka-network
    container_name: book
    \hbox{\it environment:}
      - TZ=Asia/Seoul
    build: ./book
    networks:
     - develop-net
      - develop-tool-net
     - kafka-network
  donation:
    container_name: donation
    environment:
      - TZ=Asia/Seoul
    build: ./donation
    networks:
     - develop-net
      - develop-tool-net
     - kafka-network
    container_name: chat
    \hbox{\it environment:}
      - TZ=Asia/Seoul
    build: ./chat
    networks:
     - develop-net
     - develop-tool-net
- kafka-network
  notification:
    container_name: notification
    environment:
      - TZ=Asia/Seoul
    build: ./notification
    networks:
     - develop-net
      - develop-tool-net
      - kafka-network
networks:
  {\tt develop-net:}
    name: develop-net
  develop-tool-net:
    external: true
    name: develop-tool-net
  kafka-network:
    external: true
    name: kafka-network
```

## develop-tool-net docker-compose.yml

```
version: "3"

services:
    mysql:
    container_name: mysql
    image: mysql
    ports:
        - 3306:3306
    restart: always
```

```
environment:
    - MYSQL_ROOT_PASSWORD=a308a308
    - 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
    - /exthdd/docker/develop-tool/csv:/var/lib/mysql-files
  networks:
    - develop-tool-net
mongo-db:
  container_name: mongo
  image: mongo:5.0
  restart: always
  volumes:
    - ./data:/data
  ports:
    - 27017:27017
  environment:
    {\tt MONGO\_INITDB\_ROOT\_USERNAME:} \ admin
    MONGO INITDB ROOT PASSWORD: a308a308
    MONGO_INITDB_DATABASE: chat
  networks:
    - develop-tool-net
auth-redis:
 container_name: auth-redis
  image: redis:7.0.8
  ports:
    - 6379:6379
  restart: always
    - /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
book-redis:
  container name: book-redis
  image: redis:7.0.8
  ports:
   - 6380:6379
  restart: always
  volumes:
   - /exthdd/docker/develop-tool/book-redis/conf/redis.conf:/etc/redis/redis.conf
    - /exthdd/docker/develop-tool/book-redis/data:/data
  command: redis-server /etc/redis/redis.conf
  networks:
    - develop-tool-net
  container_name: member-redis
  image: redis:7.0.8
  ports:
    - 6381:6379
  restart: always
  volumes:
    - /exthdd/docker/develop-tool/member-redis/conf/redis.conf:/etc/redis/redis.conf
    - /exthdd/docker/develop-tool/member-redis/data:/data
  command: redis-server /etc/redis/redis.conf
  networks:
    - develop-tool-net
chat-redis:
  container_name: chat-redis
  image: redis:7.0.8
  ports:
   - 6382:6379
  restart: always
  volumes:
    - /exthdd/docker/develop-tool/chat-redis/conf/redis.conf:/etc/redis/redis.conf
    - /exthdd/docker/develop-tool/chat-redis/data:/data
  command: redis-server /etc/redis/redis.conf
  networks:
    - develop-tool-net
elasticsearch:
  container_name: elasticsearch
  hostname: elasticsearch
```

```
image: docker.elastic.co/elasticsearch/elasticsearch:7.17.3
    environment:
      - node.name=es01
      - cluster.name=es-docker-cluster
      - discovery.type=single-node
     - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
    volumes:
       - /exthdd/docker/develop-tool/elasticsearch/data:/usr/share/elasticsearch/data:z
    ports:
      - 9200:9200
    networks:
      - develop-tool-net
networks:
  develop-tool-net:
    name: develop-tool-net
```

## develop-kafka-net docker-compose.yml

```
version: '3.8'
services:
  zookeeper-1:
    image: confluentinc/cp-zookeeper:5.5.1
      - '32181:32181'
    environment:
     ZOOKEEPER_CLIENT_PORT: 32181
     ZOOKEEPER_TICK_TIME: 2000
 kafka-1:
   image: confluentinc/cp-kafka:5.5.1
   ports:
     - '9092:9092'
   depends_on:
      - zookeeper-1
    environment:
     KAFKA_BROKER_ID: 1
     KAFKA_ZOOKEEPER_CONNECT: zookeeper-1:32181
     KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT, EXTERNAL:PLAINTEXT
     KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
     KAFKA_ADVERTISED_LISTENERS: INTERNAL://kafka-1:29092,EXTERNAL://k9a308.p.ssafy.io:9092
      KAFKA_DEFAULT_REPLICATION_FACTOR: 3
     KAFKA_NUM_PARTITIONS: 3
   image: confluentinc/cp-kafka:5.5.1
    ports:
      - '9093:9093'
    depends_on:
      - zookeeper-1
    environment:
     KAFKA_BROKER_ID: 2
     KAFKA_ZOOKEEPER_CONNECT: zookeeper-1:32181
     KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT, EXTERNAL:PLAINTEXT
     KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
     KAFKA_ADVERTISED_LISTENERS: INTERNAL://kafka-2:29093,EXTERNAL://k9a308.p.ssafy.io:9093
     KAFKA_DEFAULT_REPLICATION_FACTOR: 3
     KAFKA_NUM_PARTITIONS: 3
  kafka-3:
   image: confluentinc/cp-kafka:5.5.1
    ports:
      - '9094:9094'
    depends_on:
      - zookeeper-1
    environment:
     KAFKA BROKER ID: 3
      KAFKA ZOOKEEPER CONNECT: zookeeper-1:32181
     KAFKA LISTENER SECURITY PROTOCOL MAP: INTERNAL:PLAINTEXT.EXTERNAL:PLAINTEXT
     KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
     KAFKA_ADVERTISED_LISTENERS: INTERNAL://kafka-3:29094,EXTERNAL://k9a308.p.ssafy.io:9094
      KAFKA_DEFAULT_REPLICATION_FACTOR: 3
      KAFKA_NUM_PARTITIONS: 3
```

## docker-compose-ui.yml

```
services:
kafka-ui:
image: provectuslabs/kafka-ui
container_name: kafka-ui
ports:
- "8989:8080"
restart: always
environment:
- KAFKA_CLUSTERS_0_NAME=local
- KAFKA_CLUSTERS_0_BOOTSTRAPSERVERS=kafka-1:29092, kafka-2:29093, kafka-3:29094
- KAFKA_CLUSTERS_0_ZOOKEEPER=zookeeper-1:22181
```

## application.yml

## discovery

```
server:
port: 8761

spring:
application:
name: discovery-service

eureka:
client:
register-with-eureka: false
fetch-registry: false
```

### gateway

```
port: 8000
eureka:
 client:
   register-with-eureka: true
    fetch-registry: true
    service-url:
     defaultZone: http://k9a308.p.ssafy.io:8761/eureka
spring:
 application:
   name: gateway
  cloud:
    gateway:
     routes:
       - id: member-service
          uri: lb://MEMBER-SERVICE
          predicates:
            - Path=/api/members/**, /api/ranks/**
          filters:
            - CustomFilter
            - name: LoggingFilter
             args:
               baseMessage: Hi, there
               preLogger: true
       postLogger: true
- id: book-service
          uri: lb://BOOK-SERVICE
          predicates:
            - Path=/api/books/**
          filters:
            - CustomFilter
        - id: donation-service
          uri: lb://DONATION-SERVICE
             - Path=/api/donations/**, /api/trades/**, /api/histories/**
          filters:
            - CustomFilter
        - id: chat-service
```

```
uri: lb://CHAT-SERVICE
                                        predicates:
                                                 - Path=/api/chats/**, /ws/**, /app/**, /sub/**
                                        filters:
                                                - CustomFilter
                               - id: notification-service
                                       uri: lb://NOTIFICATION-SERVICE
                                        predicates:
                                                 - Path=/api/notifications/**
                                        filters:
                                                - CustomFilter
                                - id: auth-service
                                        uri: lb://AUTH-SERVICE
                                        predicates:
                                                    - Path=/api/auth/**
                                        filters:
                        default-filters:
                                  - name: GlobalFilter
                                        args:
                                               baseMessage: Spring Cloud Gateway Global Filter
                                               preLogger: true
                                               postLogger: true
jwt:
        secret:\ 2fa0936e7f2b67ca5db8472f80b822534718c8c124215f4b305ab766bc2c521780dff914130dacfaf8e9497fcf4f50e24737e593f4be82c7c7f5fbe8f6fee24f36f6fee24f36f6fee24f36f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6f6fee24f6ff6fee24f6f6fee24f6ff6fee24f6ff6fee24f6ff6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24f6fee24ffee24ffee24f6fee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ffee24ff
        access-expiration: 180000000 # 3000?
        refresh-expiration: 1209600000 # 14?
```

### auth

```
server:
 port: 0
spring:
 backend:
   host: http://k9a308.p.ssafy.io:8000
 application:
   name: auth-service
 security:
   oauth2:
     client:
       registration:
         kakao:
          client-id: #native app key # 윤혜민
           scope: profile_nickname, account_email # ??? ??
           redirect-uri: http://k9a308.p.ssafy.io:8000/api/auth/kakao # redirect ??
          authorization\hbox{-}grant\hbox{-}type\hbox{: }authorization\hbox{\_}code
          client-name: kakao
          client-authentication-method: POST
       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
   instance-id: $\{spring.application.name\}: \$\{spring.application.instance\_id: \$\{random.value\}\}
  client:
   register-with-eureka: true
   fetch-registry: true
   service-url:
     defaultZone: http://k9a308.p.ssafy.io:8761/eureka
jwt:
 access-expiration: 180000000 # 30?
 refresh-expiration: 1209600000 # 14?
redis:
 host: k9a308.p.ssafy.io
```

```
port: 6379
password: a308a308
```

### member

```
server:
     port: 0
spring:
      application:
             name: member-service
      datasource:
            \label{lem:constraint} driver-class-name: com.mysql.cj.jdbc.Driver \\ url: jdbc:mysql://k9a308.p.ssafy.io:3306/bookdone_member?serverTimezone=UTC\&characterEncoding=UTF-8\&collation=utf8mb4\_bin \\ driver-class-name: com.mysql.cj.jdbc.Driver \\ url: jdbc:mysql://k9a308.p.ssafy.io:3306/bookdone_member?serverTimezone=UTC\&characterEncoding=UTF-8\&collation=utf8mb4\_bin \\ driver-class-name: com.mysql.cj.jdbc.Driver \\ url: jdbc:mysql://k9a308.p.ssafy.io:3306/bookdone_member?serverTimezone=UTC\&characterEncoding=UTF-8\&collation=utf8mb4\_bin \\ driver-class-name: com.mysql.cj.driver \\ driver-class-name: com.mysql.cj.driver-class-name: com
             username: root
             password: a308a308
      jpa:
            database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
             open-in-view: false
             hibernate:
                   ddl-auto: update
                    show_sql: true
                    format_sql: true
                   use_sql_comments: true
       kafka:
             admin:
                  operations-timeout: 30s
             bootstrap-servers: k9a308.p.ssafy.io:9092,k9a308.p.ssafy.io:9093,k9a308.p.ssafy.io:9094
             consumer:
                  auto-offset-reset: earliest
       redis:
             port: 6381
             host: k9a308.p.ssafy.io
             password: a308a308
eureka:
            instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
            register-with-eureka: true
              fetch-registry: true
             service-url:
                    defaultZone: http://k9a308.p.ssafy.io:8761/eureka
```

### book

```
server:
      port: 0
 spring:
        application:
                name: book-service
        datasource:
                {\tt driver\text{-}class\text{-}name: com.mysql.cj.jdbc.Driver}
                 wrl: jdbc: mysql: //k9a308.p.ssafy.io: 3306/bookdone\_book? server Timezone = UTC\& character Encoding = UTF-8\& collation = utf8mb4\_bin = utf8
                username: root
                 password: a308a308
        jpa:
                database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
                 open-in-view: false
                 hibernate:
                         ddl-auto: update
                         show_sql: true
                          format_sql: true
                          use_sql_comments: true
                 defer-datasource-initialization: true
         sql:
                 init:
                          mode: always
         redis:
               host: k9a308.p.ssafy.io
                 port: 6380
                password: a308a308
         elasticsearch:
```

```
host: k9a308.p.ssafy.io
port: 9200

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://k9a308.p.ssafy.io:8761/eureka
```

## donation

```
server:
    port: 0
spring:
    application:
         name: donation-service
          driver-class-name: com.mysql.cj.jdbc.Driver
           url: jdbc: \verb|mysql|://k9a308.p.ssafy.io:3306/bookdone_donation| 2 use SSL = false \& allow Public Key Retrieval = true \& server Timezone = Asia/Seoul Leading False & Asia/Seoul Leadi
          username: root
          password: a308a308
      jackson:
          time-zone: Asia/Seoul
          {\tt database\text{-}platform:\ org.hibernate.dialect.MySQL5InnoDBDialect}
          open-in-view: false
          hibernate:
               ddl-auto: update
               show_sql: true
                format sql: true
               use_sql_comments: true
      kafka:
          admin:
               operations-timeout: 30s
          bootstrap-servers: k9a308.p.ssafy.io:9092,k9a308.p.ssafy.io:9093,k9a308.p.ssafy.io:9094
               auto-offset-reset: earliest
          producer:
               acks: all # (0, 1, -1(all) )
               res-req:
                  name: dona-req
               don-can:
                   name: dona-can
                don-fin:
                  name: dona-com
     servlet:
          multipart:
                file-size-threshold: 1MB
               location: C:/Temp
               max-file-size: 100MB
               max-request-size: 100MB
cloud:
          s3.bucket: bookdone
          stack.auto: false
          region.static: ap-northeast-2
          credentials:
               accessKey: ## key
               secretKey: ## key
     instance:
          instance-id: $\{spring.application.name\}: \$\{spring.application.instance\_id: \$\{random.value\}\} \}
      client:
          register-with-eureka: true
           fetch-registry: true
          service-url:
                defaultZone: http://k9a308.p.ssafy.io:8761/eureka
```

#### Chat

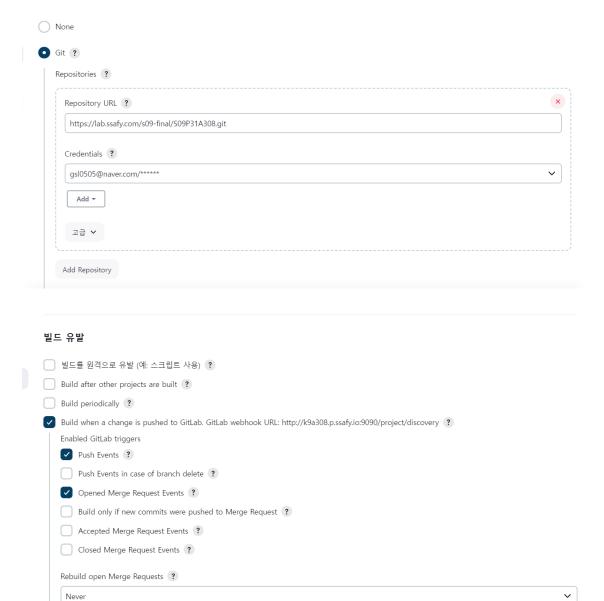
```
server:
 port: 0
sprina:
 application:
   name: chat-service
   mongodb:
     uri: mongodb://admin:a308a308@k9a308.p.ssafy.io:27017/chat
   port: 6382
   host: k9a308.p.ssafy.io
   password: a308a308
eureka:
 instance:
   \verb|instance-id: \$\{spring.application.instance\_id:\$\{random.value\}\}|
 client:
   register-with-eureka: true
    fetch-registry: true
   service-url:
     defaultZone: http://k9a308.p.ssafy.io:8761/eureka
```

### notification

```
server:
     port: 0
 spring:
     application:
            name: notification-service
      datasource:
            {\tt driver\text{-}class\text{-}name: com.mysql.cj.jdbc.Driver}
             url: jdbc: mysql://k9a308.p.ssafy.io: 3306/bookdone\_notification? server Timezone = UTC\&character Encoding = UTF-8\&collation = utf8mb4\_bin = UTC\&character =
            username: root
             password: a308a308
            database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
             open-in-view: false
             hibernate:
                  ddl-auto: none
                   show_sql: true
                   format_sql: true
                    use_sql_comments: true
      kafka:
             admin:
             bootstrap-servers: k9a308.p.ssafy.io:9092, k9a308.p.ssafy.io:9093, k9a308.p.ssafy.io:9094
             consumer:
                  auto-offset-reset: earliest
firebase:
      sdk-path: firebase/bookdone-3a0d1-firebase-adminsdk-llxqi-754ac11e18.json
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://k9a308.p.ssafy.io:8761/eureka
```

# pipeline

### 소스 코드 관리



### **Build Steps**



#### 빌드 후 조치

