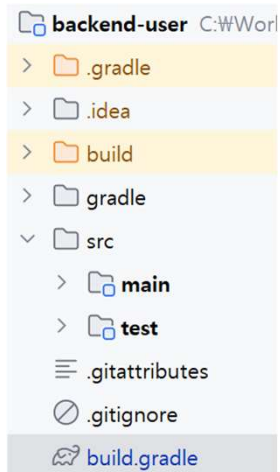


Spring Cloud Config Client 설정

프로젝트 의존성 추가 : gateway, eureka, user, post, alim 대상



```
dependencies {
```

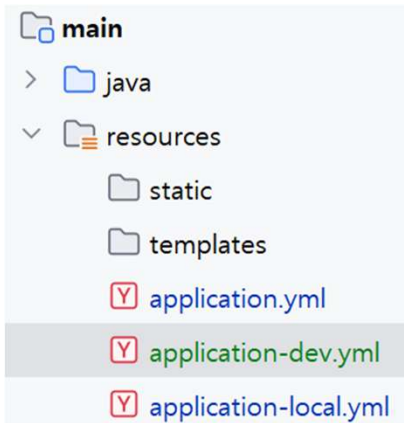
```
...
```

```
implementation 'org.springframework.cloud:spring-cloud-starter-config'
```

```
...
```

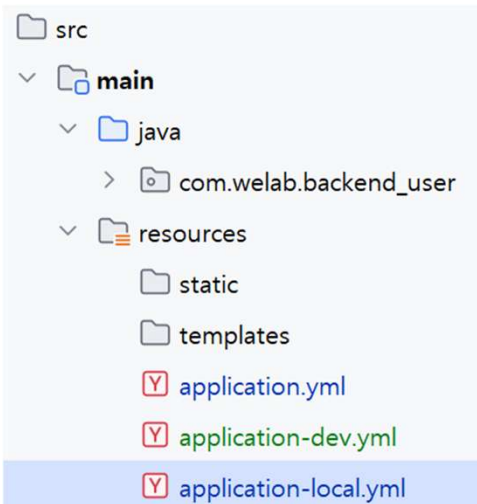
```
}
```

application-dev.yml 추가 : gateway, eureka, user, post, alim 대상



```
spring:  
  config:  
    import: optional:configserver:http://host.docker.internal:8888
```

application-local.yml 추가 : gateway, eureka, user, post, alim 대상



```
spring:
  ...
zipkin:
  tracing:
    endpoint: http://localhost:9411/api/v2/spans
cloud:
  config:
    enabled: false
```

Dockerfile 추가 : gateway, eureka, user, post, alim 대상

- ≡ .gitattributes
- 🚫 .gitignore
- 🐙 build.gradle
- 🐳 Dockerfile**
- 📄 gradlew
- ≡ gradlew.bat
- M↓ HELP.md

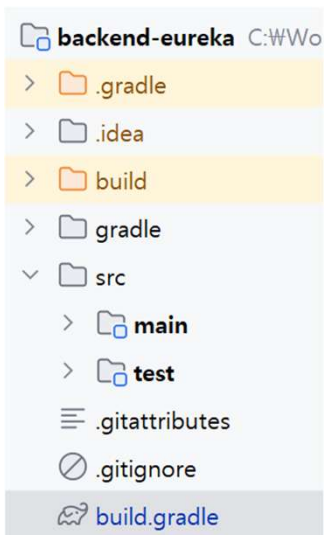
```
FROM amazoncorretto:17
MAINTAINER dev@welab.com
VOLUME /tmp
EXPOSE 8080
COPY build/libs/*.jar /app.jar
ENTRYPOINT ["java", "-Djava.security.egd=file:/dev/./urandom", "-jar", "/app.jar"]
```

EXPOSE

8080 : User, Post, Alim, Gateway

8761 : Eureka

Eureka 프로젝트 의존성 추가



```
dependencies {  
    implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-server'  
    implementation 'org.springframework.cloud:spring-cloud-starter-config'  
    implementation 'org.springframework.cloud:spring-cloud-starter-bootstrap'  
    implementation 'org.springframework.boot:spring-boot-starter-actuator'  
    compileOnly 'org.projectlombok:lombok'  
    annotationProcessor 'org.projectlombok:lombok'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
    testRuntimeOnly 'org.junit.platform:junit-platform-launcher'  
}
```

가

각 앱 Build & Release

```
PS C:\Workspace\spring-cloud\backend-config> ./gradlew clean build
```

```
PS C:\Workspace\spring-cloud\backend-config> docker build -t solarhc/backend-user:0.0.1 .
```

```
PS C:\Workspace\spring-cloud\backend-config> docker push solarhc/backend-user:0.0.1
```

Diagram illustrating the components of the Docker push command:

- `solarhc`: docker hub id
- `backend-user`: app name
- `0.0.1`: version

Backend service 실행

노션에 있는 *docker-compose.yml* 파일을 아래 디렉토리에 위치

C:\server\service>

docker compose up -d 명령어 실행

C:\server\service> **docker compose up -d**