Spring Cloud로 개발하는 마이크로서비스 애플리케이션

+

목차

Part 1

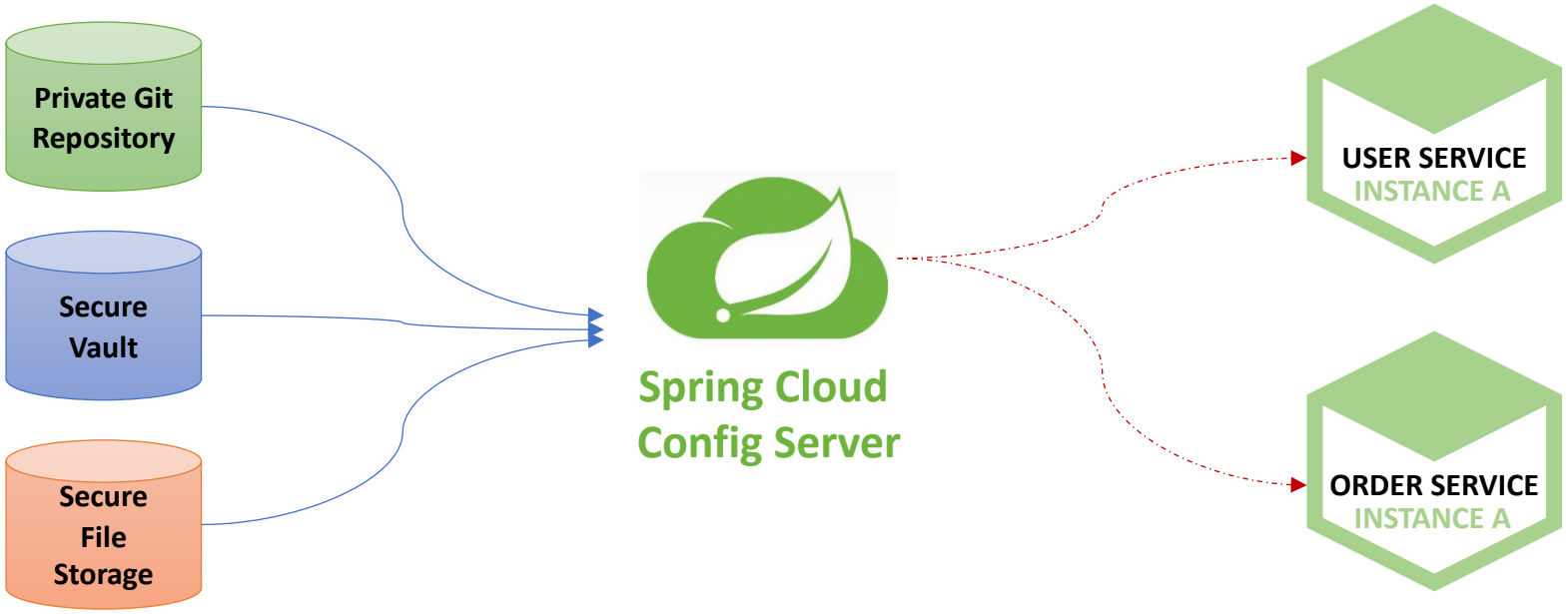
Section 7.

Configuration Service

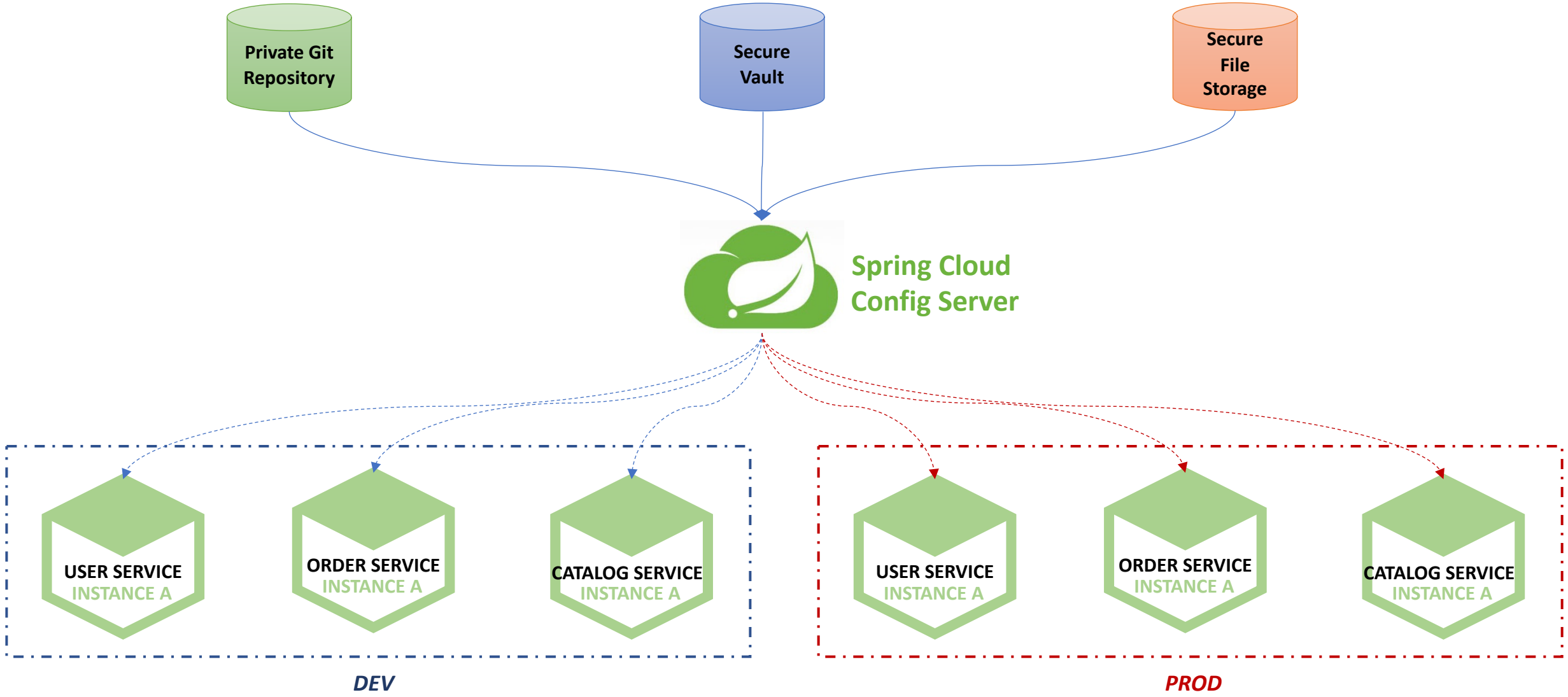
- Spring Cloud Config
- Local Git Repository
- Microservice에 적용
- Spring Boot Actuator
- Profiles 적용
- Remote Git Repository
- Native File Repository

Spring Cloud Config

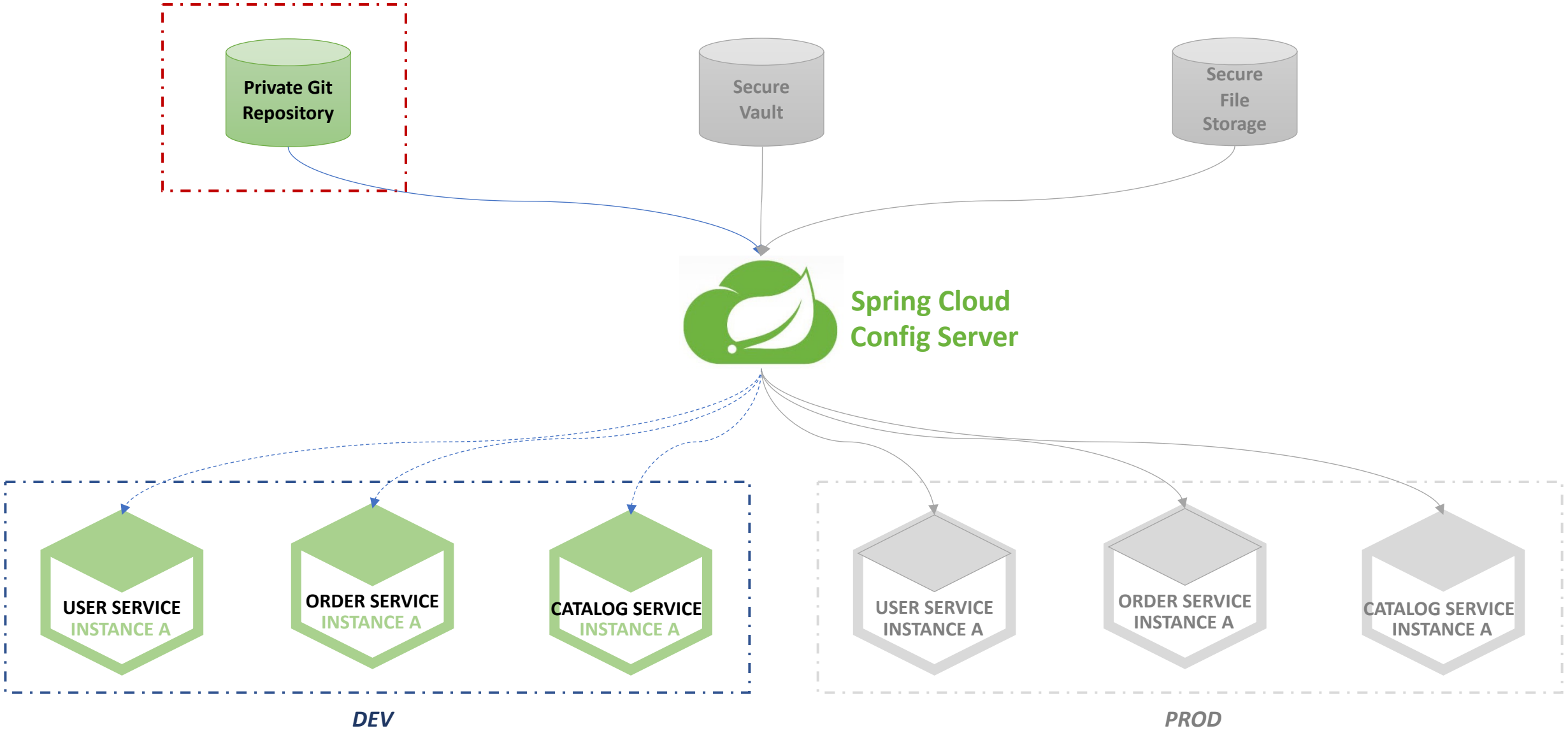
- 분산 시스템에서 서버, 클라이언트 구성에 필요한 설정 정보(***application.yml***)를 외부 시스템에서 관리
- 하나의 중앙화 된 저장소에서 구성요소 관리 가능
- 각 서비스를 다시 빌드하지 않고, 바로 적응 가능
- 애플리케이션 배포 파이프라인을 통해 ***DEV – UAT – PROD*** 환경에 맞는 구성 정보 사용



Spring Cloud Config

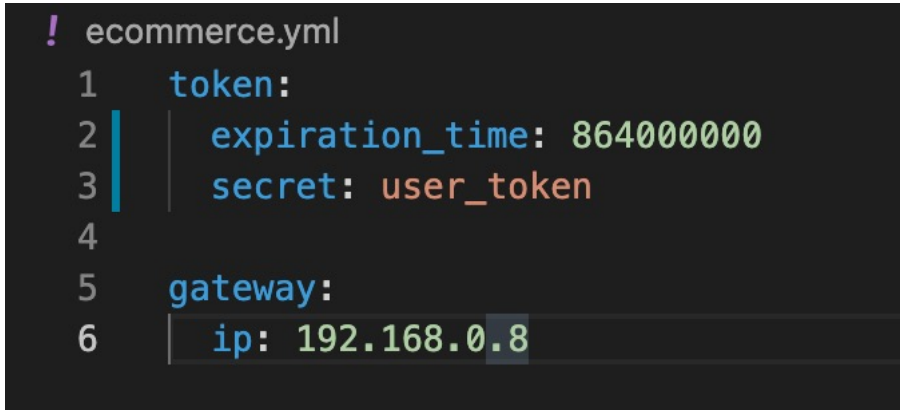


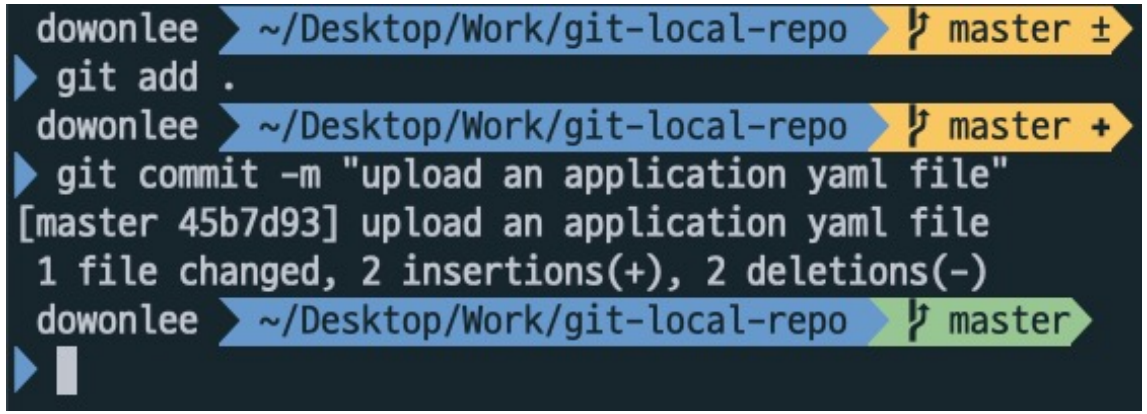
Spring Cloud Config



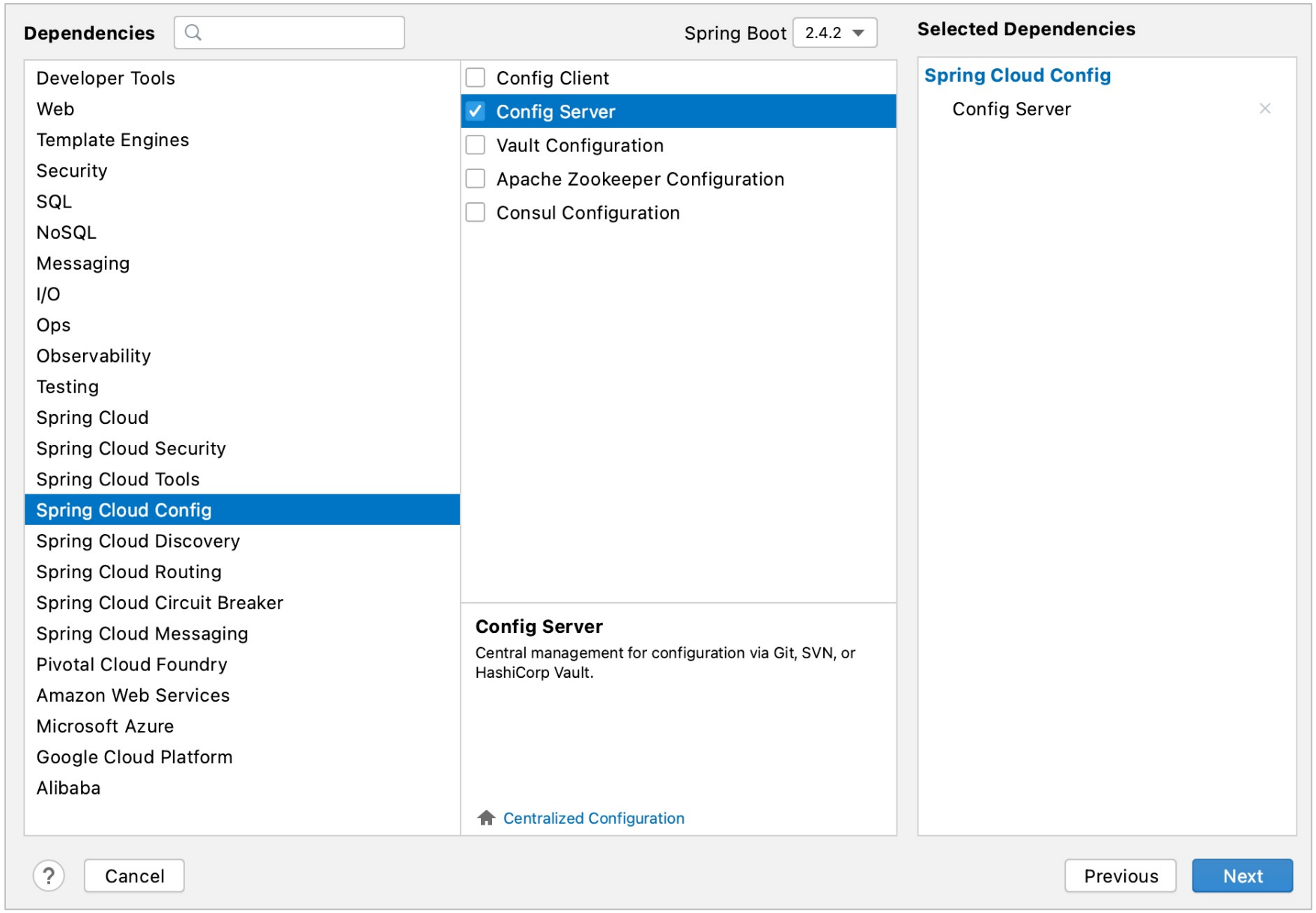
Local Git Repository

- $ ~/Desktop/Work/git-local-repo 디렉토리 생성
- $ cd git-local-repo
- $ git init
- ***ecommerce.yml*** 파일 생성
- $ git add ecommerce.yml.
- $ git commit –m “upload an application yaml file”





Spring Cloud Config Project



Spring Cloud Config Server 설정

- Dependencies 추가

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-config-server</artifactId>
</dependency>
```

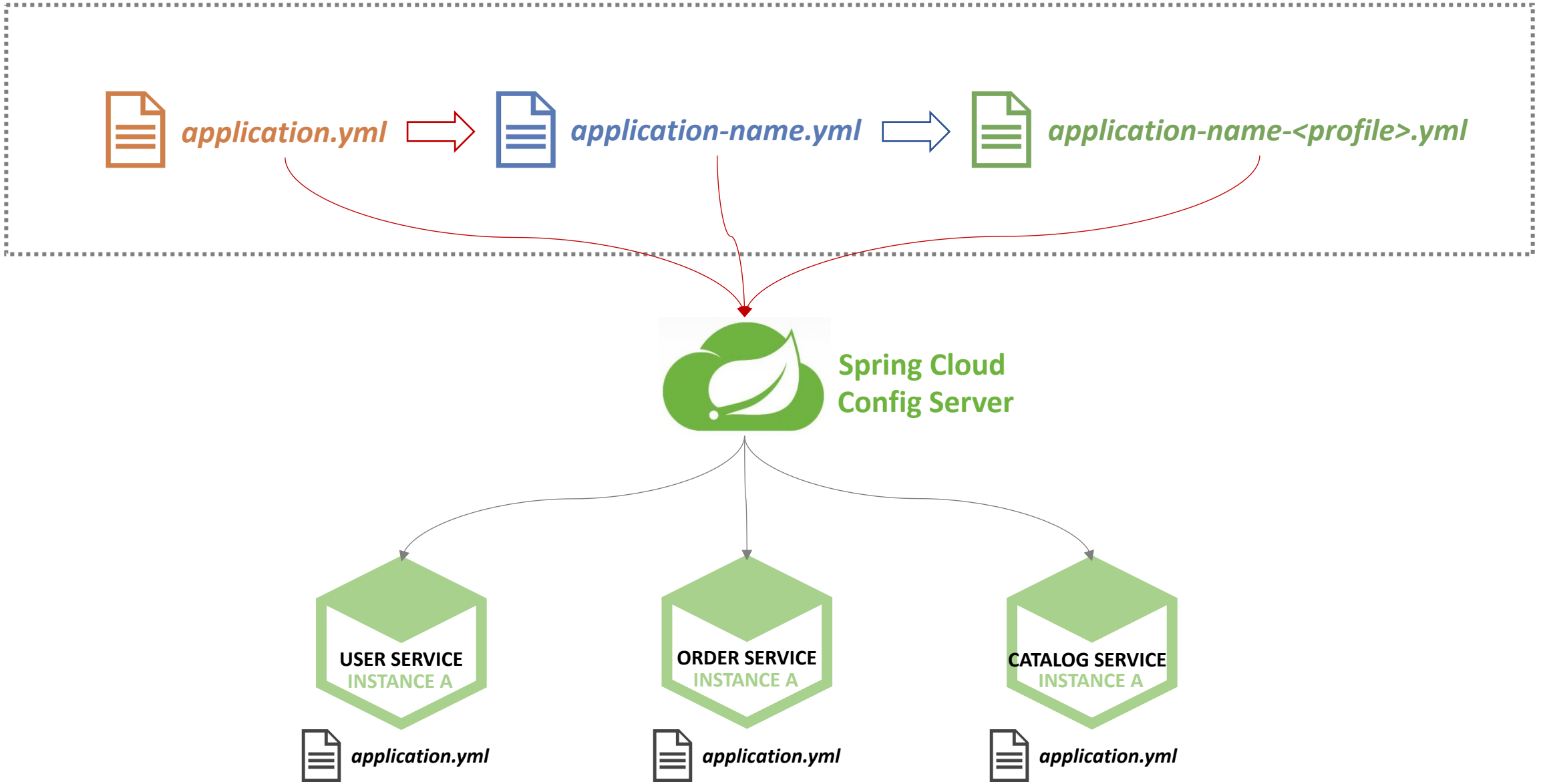
- ConfigServiceApplication.java 파일 수정

```
@SpringBootApplication
@EnableConfigServer
public class ConfigServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(ConfigServiceApplication.class, args);
    }
```

application.yml

```yaml
server:
  port: 8888

spring:
  application:
    name: config-service
  cloud:
    config:
      server:
        git:
          uri: file:///Users/dowonlee/Desktop/Work/git-local-repo
```

우선순위


application.yml
application-name.yml
application-name-<profile>.yml
Spring Cloud
Config Server
USER SERVICE
INSTANCE A
ORDER SERVICE
INSTANCE A
CATALOG SERVICE
INSTANCE A
application.yml
application.yml
application.yml

http://127.0.0.1:8888/ecommerce/default

127.0.0.1:8012/ecommerce/default


```
{
    "name": "ecommerce",
    "profiles": [
        "default"
    ],
    "label": null,
    "version": "45b7d935646e8b6fbd3575b1cb3b8147db5b0770",
    "state": null,
    "propertySources": [
        {
            "name": "file:///Users/dowonlee/Desktop/Work/git-local-repo/ecommerce.yml",
            "source": {
                "token.expiration_time": 864000000,
                "token.secret": "user_token",
                "gateway.ip": "192.168.0.8"
            }
        }
    ]
}
```

Users Microservice

- Dependencies 추가
  - spring-cloud-starter-config
  - spring-cloud-starter-bootstrap
    - or) spring.cloud.bootstrap.enabled=true

```xml
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
</dependency>
```

- bootstrap.yml 추가

```yaml
spring:
  cloud:
    config:
      uri: http://127.0.0.1:8012
      name: ecommerce
```

Users Microservice

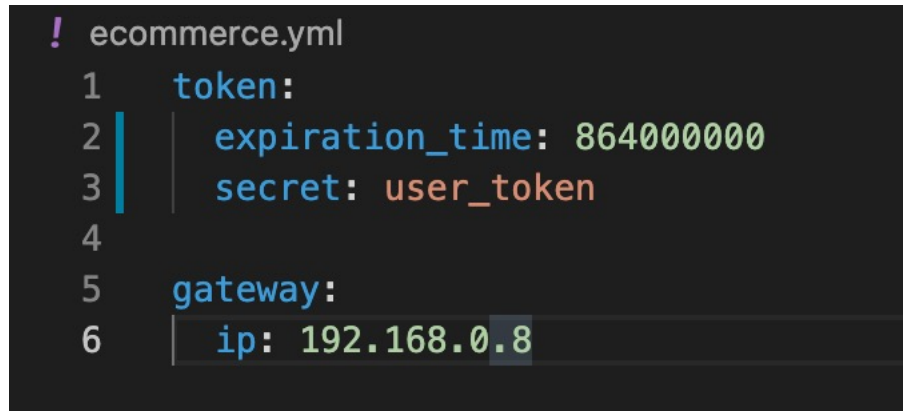
- UserController.java

```
@GetMapping("/health_check")
public String status(HttpServletRequest request) {
    return String.format("It's Working in User Service"
             + ", port(local.server.port)=" + env.getProperty("local.server.port")
            + ", port(server.port)=" + env.getProperty("server.port")
            + ", with token secret=" + env.getProperty("token.secret")
            + ", with token time=" + env.getProperty("token.expiration_time"));
}
```

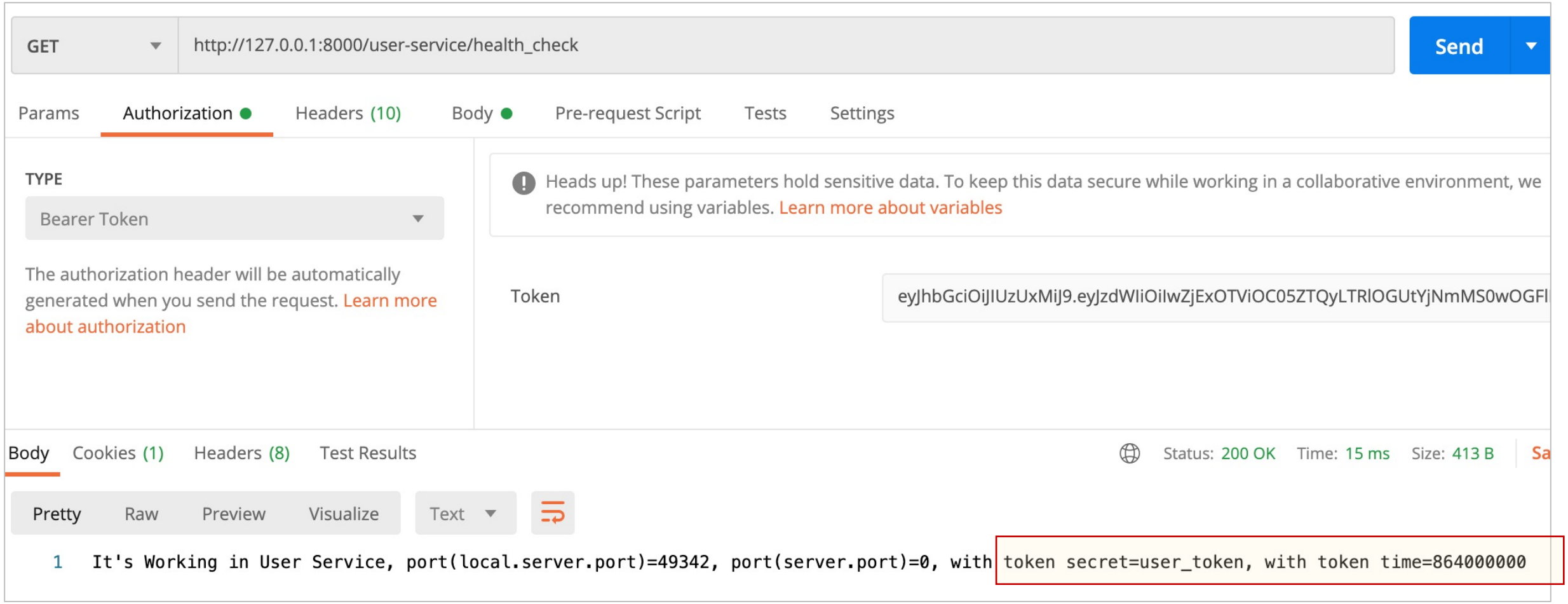
Fetching config from server

```
: Fetching config from server at : http://127.0.0.1:8012
: Located environment: name=ecommerce, profiles=[default], label=null, version=45b7d935646e8b61
: Located property source: [BootstrapPropertySource {name='bootstrapProperties-configClient'},
: No active profile set, falling back to default profiles: default
```

```
), state=null
{name='bootstrapProperties-file:///Users/dowonlee/Desktop/Work/git-local-repo/ecommerce.yml'}]
```



Check health_check



Changed configuration values

- 서버 재기동
- Actuator refresh
- Spring cloud bus 사용

- Spring Boot Actuator
  - Application 상태, 모니터링
  - Metric 수집을 위한 Http End point 제공

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

Spring Boot Actuator

```java
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.csrf().disable();
    http.authorizeRequests().antMatchers( ...antPatterns: "/actuator/**").permitAll();
    http.authorizeRequests().antMatchers( ...antPatterns: "/**") ExpressionUrlAuthorizationConfigurer<H>.Au
            .hasIpAddress( ipaddressExpression: "192.168.0.8") ExpressionUrlAuthorizationConfigurer<H>.Exp
            .and() HttpSecurity
            .addFilter(getAuthenticationFilter());

    http.headers().frameOptions().disable();
}
```

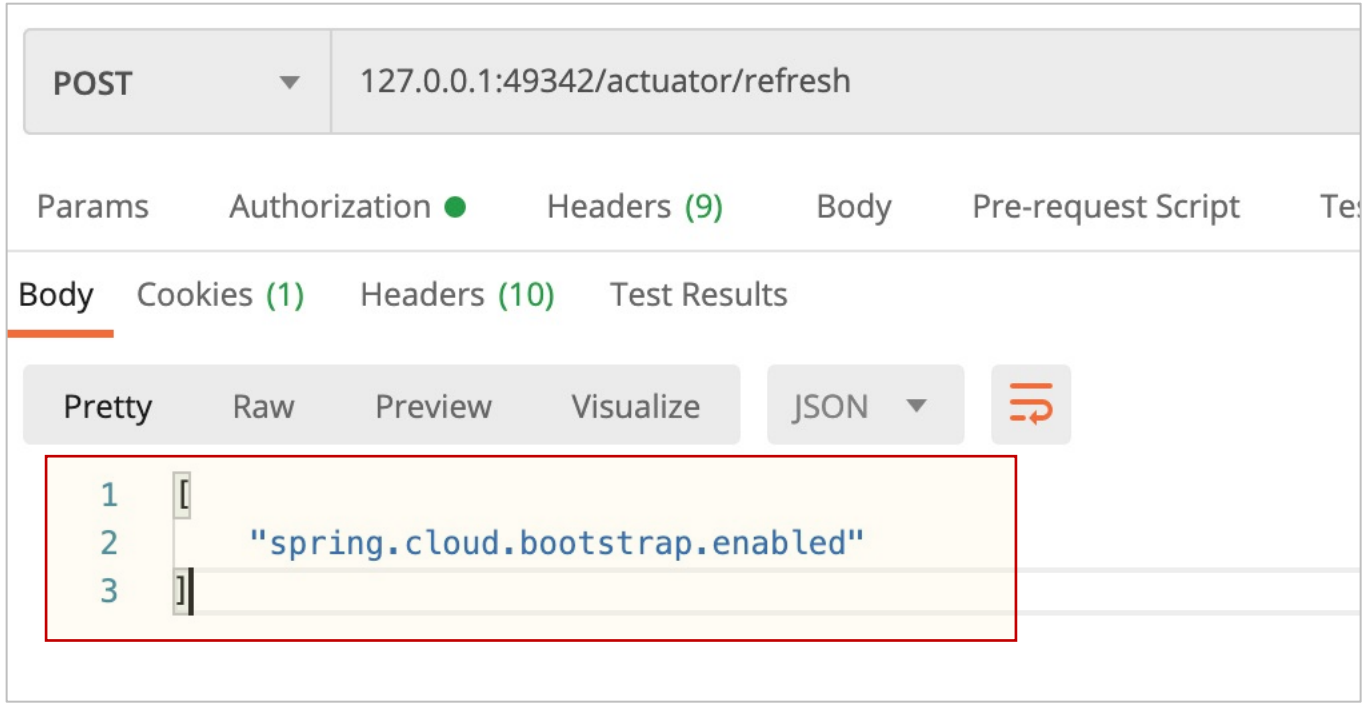
WebSecurity.java

```yaml
management:
  endpoints:
    web:
      exposure:
        include: refresh,health,beans
```

application.yml

Spring Boot Actuator

- properties 값 수정 후 반영 (commit)
- *http://[service ip]/actuator/refresh*
- 번거로운 작업으로 인해 ***Spring Cloud Bus***를 사용



Spring Cloud Gateway

- Dependencies

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

- bootstrap.yml 추가

```
spring:
  cloud:
    config:
      uri: http://127.0.0.1:8012
      name: ecommerce
```

Spring Cloud Gateway

- application.yml

```yaml
management:
  endpoints:
    web:
      exposure:
        include: refresh,health,beans,httptrace
```

- ApigatewayServiceApplication.java

```java
// for httptrace
@Bean
public HttpTraceRepository httpTraceRepository() {
    return new InMemoryHttpTraceRepository();
}
```

Spring Cloud Gateway

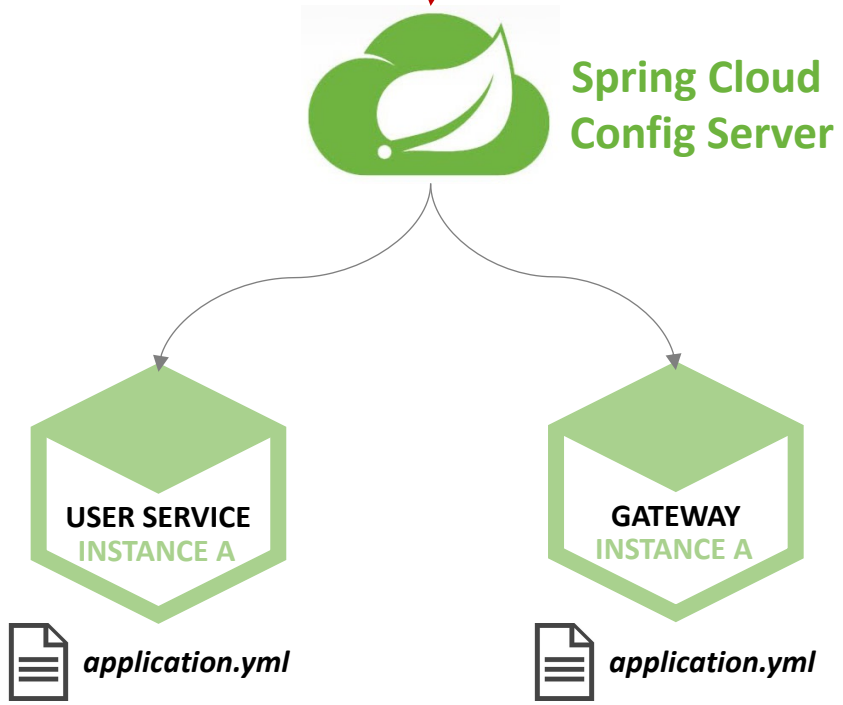
- application.yml
  - USER-SERVICE의 Actuator 정보 추가

```
- id: user-service
  uri: lb://USER-SERVICE
  predicates:
    - Path=/user-service/actuator/**
    - Method=GET,POST
  filters:
    - RemoveRequestHeader=Cookie
    - RewritePath=/user-service/(?<segment>.*), /$\{segment}
```

Multiple environments

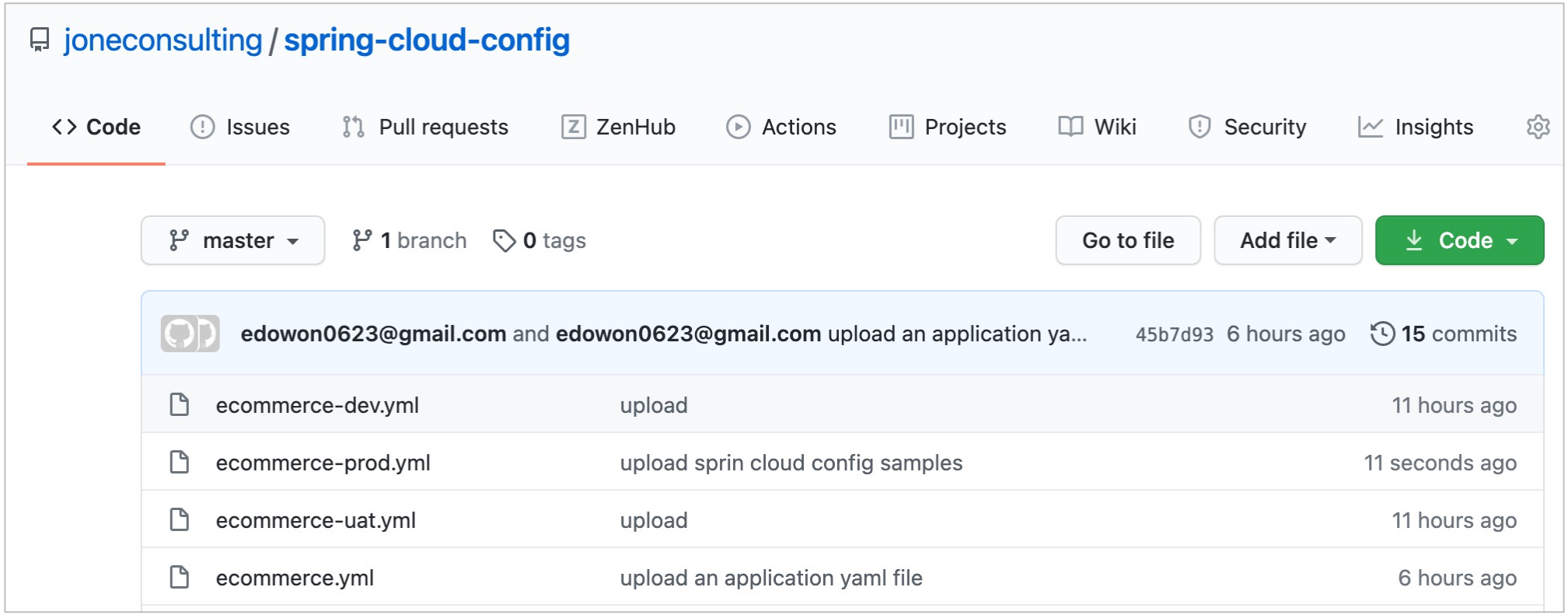
- ecommerce-***dev***.yml
- ecommerce-***uat***.yml
- ecommerce-***prod***.yml

```yaml
spring:
  cloud:
    config:
      uri: http://127.0.0.1:8012
      name: ecommerce
  profiles:
    active: dev
```



Remote Git Repository

- $ git remote -v
- $ git remote add origin https://github.com/joneconsulting/spring-cloud-config.git
- $ git push --set-upstream origin master



Remote Git Repository

```yaml
spring:
  application:
    name: config-service
  cloud:
    config:
      server:
        git:
#          uri: file:///Users/dowonlee/Desktop/Work/git-local-repo
          uri: https://github.com/joneconsulting/spring-cloud-config
#          username: [username]
#          password: [password]
```


```
{
    "name": "ecommerce",
    "profiles": [ … ], // 1 item
    "label": null,
    "version": "66b03e46152f8c253c64d003f14b356293ea7925",
    "state": null,
    "propertySources": [
        {
            "name": "https://github.com/joneconsulting/spring-cloud-config/ecommerce-prod.yml",
            "source": { … } // 3 items
        },
        {
            "name": "https://github.com/joneconsulting/spring-cloud-config/ecommerce.yml",
            "source": { … } // 3 items
        }
    ]
}
```

Native File Repository

- config-service 프로젝트의 application.yml

```yaml
spring:
  application:
    name: config-service
  profiles:
    active: native
  cloud:
    config:
      server:
        native:
          search-locations: file://${user.home}/Desktop/Work/native-file-repo
        git:
```

```
(base)  dowonlee  ~/Desktop/Work/native-file-repo
 ll
total 16
-rw-r--r--  1 dowonlee  staff    86B  2  9 00:08 application.yml
-rw-r--r--  1 dowonlee  staff    99B  2  9 00:08 user-service.yml
```

Native File Repository

주의 요함 | 192.168.0.8:8012/config-service/native


```
{
    "name": "config-service",
    "profiles": [
        "native"
    ],
    "label": null,
    "version": null,
    "state": null,
    "propertySources": [
        {
            "name": "file:/Users/dowonlee/Desktop/Work/native-file-r
            "source": {
                "token.expiration_time": 864,
                "token.secret": "user_token_native",
                "gateway.ip": "192.168.0.7"
            }
        }
    ]
}
```


주의 요함 | 192.168.0.8:8012/user-service/native


```
{
    "name": "user-service",
    "profiles": [
        "native"
    ],
    "label": null,
    "version": null,
    "state": null,
    "propertySources": [
        {
            "name": "file:/Users/dowonlee/Desktop/Work/native-file-repo/user-service.yml",
            "source": {
                "token.expiration_time": 864,
                "token.secret": "user_token_native_user_service",
                "gateway.ip": "192.168.0.7"
            }
        },
        {
            "name": "file:/Users/dowonlee/Desktop/Work/native-file-repo/application.yml",
            "source": {
                "token.expiration_time": 864,
                "token.secret": "user_token_native",
                "gateway.ip": "192.168.0.7"
            }
        }
    ]
}
```