您太敏锐了，完全抓住了问题的核心！

是的，**pom.xml 文件中的 <version>1.5.3.RELEASE</version> 正是导致您之前一系列问题的根源**。

这是一个非常古老的 Spring Boot 版本（大约发布于2017年），与之配套的 Spring Cloud 版本 (Dalston.RELEASE) 也同样过时了。使用这些旧版本会导致以下问题：

* **依赖冲突**: 无法与现代的库（如新的 Actuator）兼容，导致 Unresolved dependency 错误。
* **功能缺失**: 缺少 Spring Boot 2.x 带来的大量性能优化、安全增强和新功能。
* **社区停止支持**: 遇到问题很难找到解决方案，并且存在潜在的安全漏洞。

### 解决方案：升级到现代、稳定的 Spring Boot 版本

要彻底解决这个问题，我们需要对 pom.xml 进行一次升级。我将指导您升级到一个**长期支持（LTS）且与 Java 8 兼容**的稳定版本组合：**Spring Boot 2.7.x** 和 **Spring Cloud 2021.0.x**。

### **第一步：更新 pom.xml 的核心版本**

请将您的 eureka-service 模块的 pom.xml 文件内容替换为以下更新后的版本。我已经为您标记了关键的修改点。

XML

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.techprimers.cloud</groupId>  
 <artifactId>eureka-service</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <name>eureka-service</name>  
 <description>Demo project for Spring Boot</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.18</version> <relativePath/>  
 </parent>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
 <java.version>1.8</java.version>  
 <spring-cloud.version>2021.0.8</spring-cloud.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>${spring-cloud.version}</version> <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 </dependencies>  
 </dependencyManagement>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

### **第二步：更新配置文件（application.yml 或 properties）**

从 Spring Boot 1.x 升级到 2.x，一些配置项的名称也发生了变化。请检查并更新您的 eureka-service 模块下的 application.properties 或 application.yml 文件。

这是一个推荐的、适用于 Spring Boot 2.x 的 application.yml 配置：

YAML

# eureka-service/src/main/resources/application.yml  
  
server:  
 port: 8761 # Eureka Server 端口  
  
spring:  
 application:  
 name: eureka-server  
  
eureka:  
 instance:  
 hostname: localhost  
 client:  
 # Eureka Server 不需要将自己注册为客户端  
 register-with-eureka: false  
 # Eureka Server 不需要从其他地方获取注册信息  
 fetch-registry: false  
 service-url:  
 # 为客户端提供注册地址  
 defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

**注意**：在 Spring Boot 2.x 中，默认不再需要手动排除 jersey 依赖，依赖管理更加智能。

### **第三步：重新构建和启动**

1. **保存** 修改后的 pom.xml 文件。
2. 在您的 IDE (如 IntelliJ IDEA) 中，**重新加载 Maven 项目**，让它下载新的依赖版本。
3. 执行 Maven 命令进行全新构建：  
   Bash  
   mvn clean install
4. 重新启动您的 EurekaServerApplication 服务。

完成以上步骤后，您的 Eureka Server 就会基于一个现代、稳定且受支持的 Spring Boot 版本运行。您之前遇到的 Unresolved dependency 和 Actuator 无法访问的问题都将迎刃而解。